

**Національний університет кораблебудування  
імені адмірала Макарова**

Інститут комп'ютерних наук та управління проектами  
(повне найменування інституту, назва факультету)

Кафедра інформаційних управляючих систем та технологій  
(повна назва кафедри)

**Пояснювальна записка**

до дипломного проекту (роботи)

бакалавр

(освітньо-кваліфікаційний рівень)

на тему Розробка підсистеми “Обслуговування пасажирів” ІС аеропорту

Виконав: студент 4 курсу, групи 4142  
напряму підготовки (спеціальності)

122 «Комп'ютерні науки»

(шифр і назва напряму підготовки, спеціальності)

Борис А.В.

(прізвище та ініціали)

Керівник Маршак О.І.

(прізвище та ініціали)

Рецензент Михелєв І.Л.

(прізвище та ініціали)

м. Миколаїв – 2020 року

## Анотація

Темою дипломної роботи є розробка підсистеми “Обслуговування пасажирів” ІС аеропорту. Підсистема спрямована на підвищення ефективності відділу аеропорту з обслуговування пасажирів. Основною метою інформаційної підсистеми є забезпечення зручного продажу квитків, реєстрації багажу та пасажирів на виліт, надання розкладу польотів для можливих пасажирів та диспетчерської служби.

В роботі виконано опис об’єкта, його організаційної структури, визначенні недоліки, сформовано вимоги до підсистеми, проведено аналіз існуючих рішень та розроблено концепцію інформаційної підсистеми і здійснено розробку технічного проекту. При створенні технічного проекту було здійснено розробку діаграм використання та діяльності, концептуальної, логічної та фізичної бази даних, прийнято рішення з програмного забезпечення, які є необхідними для реалізації підсистеми та розроблено діаграму розгортання.

В результаті розробки було створено підсистему “Обслуговування пасажирів” ІС аеропорту, яка представлена у вигляді веб-додатку. Додаток створений мовою програмування C# та з використанням фреймворків: ASP.NET MVC 5, Entity Framework, ASP.NET Identity.

Робота викладена на 118 сторінках друкованого тексту, містить 43 рисунка, 3 таблиці та список використаної літератури з 22 найменувань.

Ключові слова: відділ, підсистема, задача, підприємство, автоматизація, аеропорт.

					ДР.122.4142.03.ПЗ	Аркуш
						1
Зм.	Аркуш	№ документа	Підпис	Дата		

## Annotation

The theme of the work is the development of the subsystem "Passenger Service" of the airport IS. The subsystem is aimed at improving the efficiency of the airport's passenger service department. The main purpose of the information subsystem is to ensure the convenient sale of tickets, check-in of luggage and passengers on departure, providing a flight schedule for potential passengers and the dispatch service.

The paper describes the object, its organizational structure, identifies shortcomings, formed the requirements for the subsystem, analyzed the existing solutions and developed the concept of the information subsystem and developed a technical project. During the creation of the technical project, development and operation diagrams, conceptual, logical and physical database were developed, software decisions were made, which are necessary for the implementation of the subsystem, and a deployment diagram was developed.

As a result of the development, the "Passenger Service" subsystem of the airport IS was created, which is presented in the form of a web application. This application is created in the C# programming language and using frameworks: ASP, NET MVC 5, Entity Framework, ASP.NET Identity.

The work is presented on 118 pages of printed text, contains 43 pictures 3 tables and a list of used literature from 22 titles.

Keywords: department, subsystem, task, enterprise, automation, airport.

					ДР.122.4142.03.ПЗ	Аркуш
						2
Зм.	Аркуш	№ документа	Підпис	Дата		

## Зміст

ВСТУП.....	5
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ, РОЗРОБКА КОНЦЕПЦІЇ ПІДСИСТЕМИ “ОБСЛУГОВУВАННЯ ПАСАЖИРІВ” ІС АЕРОПОРТУ ТА ПОСТАНОВКА ЗАДАЧІ.....	7
1.1. Аналіз предметної області.....	8
1.1.1. Опис об’єкта автоматизації та його організаційна структура .....	8
1.1.2. Аналіз предметної області процесів обробки інформації.....	13
1.2. Формування вимог до підсистеми.....	16
1.3. Аналіз існуючих рішень .....	21
1.4. Розробка концепції підсистеми .....	24
1.4.1. Цільове призначення та задача роботи.....	24
1.4.2. Розробка функціональної моделі підсистеми .....	25
1.4.3. Формування вхідної, довідкової та вихідної інформації бази даних підсистеми .....	28
1.4.4. Вибір ОС та програмного інструментарію для реалізації підсистеми .....	32
1.4.5. Визначення засобів обчислювальної техніки.....	37
1.5. Постановка задачі.....	38
1.6. Висновки.....	41
2. РОЗРОБКА ТЕХНІЧНОГО ПРОЕКТУ ПІДСИСТЕМИ “ОБСЛУГОВУВАННЯ ПАСАЖИРІВ” ІС АЕРОПОРТУ .....	42
2.1. Загальносистемні рішення підсистеми .....	43
2.1.1. Організаційна структура.....	43
2.1.2. Функціональна структура .....	44
2.1.3. Загальний опис підсистеми.....	53
2.1.4. Діаграма використання.....	54
2.1.5. Діаграма діяльності.....	60
2.2. Рішення з інформаційного забезпечення підсистеми .....	63
2.2.1. Концептуальна та логічна моделі бази даних .....	63
2.2.2. Аналіз нормалізації сутностей бази даних:.....	66
2.3. Рішення з програмного забезпечення підсистеми .....	67
2.3.1. Структура програмного забезпечення.....	67
2.3.2. Фізична модель бази даних.....	68
2.3.3. Методи і засоби розробки програмного забезпечення .....	70

					ДР.122.4142.03.ПЗ			
Зм.	Аркаш	№ документа	Підпис	Дата				
					Розробка підсистеми “Обслуговування пасажирів” ІС аеропорту	Літ.	Аркуш	Аркушів
Студент	Борис А. В.						3	118
Керівник	Маршак О. І.					НУК		
Зав. каф.	Михелєв І. Л.							

2.4.	Діаграма розгортання .....	71
2.5.	Висновки.....	73
3.	РЕАЛІЗАЦІЯ ПІДСИСТЕМИ “ОБСЛУГОВУВАННЯ ПАСАЖИРІВ” ІС АЕРОПОРТУ .....	74
3.1.	Технічне завдання.....	75
3.2.	Опис розробки підсистеми .....	75
3.3.	Інструкція з використання підсистеми .....	76
3.4.	Висновки.....	86
4.	ОХОРОНА ПРАЦІ .....	88
4.1.	Вступ.....	89
4.2.	Аналіз шкідливих та небезпечних факторів на робочому місці .....	90
4.2.1.	Рівень штучного освітлення .....	91
4.2.2.	Мікроклімат робочого зони .....	92
4.2.3.	Шум та вібрації у робочому приміщенні .....	94
4.3.	Ергономіка робочого місця .....	95
	ВИСНОВКИ.....	100
	СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	101
	Додаток А. Технічне завдання на розробку .....	105
	Додаток Б. Скрипт створення бази даних підсистеми .....	109
	Додаток В. Лістинги модулів.....	114

## ВСТУП

Дипломна робота присвячена розробці підсистеми “Обслуговування пасажирів” ІС аеропорту.

Розробка підсистеми (ПС) «Обслуговування пасажирів» ІС аеропорту покращить функціонування роботи інформаційної системи, збільшить ефективність відділу аеропорту з обслуговування пасажирів, підвищить продуктивність роботи персоналу, покращить якість обслуговування клієнтів, автоматизує процеси обробки даних аеропорту, усуне недоліки служби з продажу квитків. реєстрації пасажирів та багажу, покращить розклад польотів для потенційних пасажирів аеропорту, що свідчить про актуальність обраної тематики.

Метою роботи є розробка підсистеми “Обслуговування пасажирів” ІС аеропорту для покращення якості обслуговування пасажирів та збільшення продуктивності роботи відділу з обслуговування пасажирів.

Основне завдання підсистеми це забезпечення якісного та зручного продажу авіаквитків клієнтам аеропорту, можливість реєстрації пасажирів та багажу на виліт, побудови розкладу вильотів та прильотів для потенційних пасажирів і диспетчерської служби, адміністрування підсистеми.

Етапи розробки підсистеми :

- характеристика об'єкта і аналізу станів – опис та характеристика об'єкта , його організаційної структури і аналізу станів та процесів обслуговування клієнтів;
- формування вимог до ПС – формування функціональних, програмних технічних вимог до ПС;
- розробка концепції ПС – розробка блок-схеми підсистеми та опис всіх служб;
- розробка технічного проекту ПС – розробка діаграм використання та діяльності, концептуальної, логічної та фізичної моделі бази даних,

					ДР.122.4142.03.ПЗ	Аркуш
						5
Зм.	Аркуш	№ документа	Підпис	Дата		

прийняття рішення, що до програмного забезпечення та розробка діаграми розгортання.

- реалізація проекту ПС – реалізація підсистеми, розробка інструкції використання підсистеми.

Підсистема призначена для клієнтів, працівників аеропорту (диспетчерів, адміністраторів касирів, працівників з реєстрації, економістів та інших співпрацівників відділу обслуговування пасажирів).

Результатом виконання дипломної роботи є підсистема «Обслуговування пасажирів» ІС аеропорт, яка зберігає необхідну інформацію в БД, та її програмний інтерфейс.

З цього, можна зробити підсумок, що аеропорт чекає нова хвиля у розвитку, оскільки впровадження підсистеми покращить якість роботи всього підприємства, що відобразиться на збільшенні прибутку і конкурентоспроможності аеропорту.

					ДР.122.4142.03.ПЗ	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		6

					ДР.122.4142.03.ПЗ.Р01				
Зм.	Аркаш	№ документа	Підпис	Дата					
					Аналіз предметної області, розробка концепції підсистеми “Обслуговування пасажирів” ІС аеропорту	Літ.	Аркуш	Аркушів	
Студент	Борис А. В.							7	35
Керівник	Маршак О. І.					НУК			
Зав. каф.	Михелєв І. Л.								



# 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ, РОЗРОБКА КОНЦЕПЦІЇ ПІДСИСТЕМИ “ОБСЛУГОВУВАННЯ ПАСАЖИРІВ” ІС АЕРОПОРТУ ТА ПОСТАНОВКА ЗАДАЧІ

## 1.1. Аналіз предметної області

### 1.1.1. Опис об’єкта автоматизації та його організаційна структура

Підприємство “Аеропорт” є комерційним підприємством цивільної авіації, яке засноване на державній власності та входить до сфери управління Міністерства інфраструктури України (Уповноважений орган управління). 100% “Аеропорт” належить Державі в особі Міністерства інфраструктури України [8].

Головними завданнями підприємства “Аеропорт” є:

- отримання прибутку від здійснення господарської діяльності;
- своєчасне задоволення попиту економіки та суспільних потреб в наданні послуг для здійснення авіаційних перевезень;
- забезпечення авіаційної безпеки та безпеки польотів.

Підприємство є суб’єктом природних монополій в частині забезпечення посадки-зльоту повітряних суден, забезпечення авіаційної безпеки, забезпечення наднормативної стоянки повітряного судна, забезпечення комунальних послуг на території аеропорту.

Завдяки активній політиці залучення авіаперевізників, до аеропорту виконують польоти більше 40 авіакомпаній, серед яких: Air Arabia, Air Astana, Air Baltic, Air France, Air Malta, Air Moldova, Adria Airways, Atlasjet Ukraine, Azerbaijan Hava Yollary, Azur Air, Austrian Airlines, Belavia, British Airways, Bravo Airways, Brussels Airlines, Bukovyna, Czech Airlines, El Al, Ellinair, Flydubai, Georgian Airways, Iraqi Airways, KLM, LOT, Lufthansa, Myway Airlines, Qatar Airways, Ryanair, SkyUp, SWISS, Turkish Airlines, Ukraine International Airlines, Wind Rose, YanAir.

Це один із аеропортів, що успішно конкурує з великими європейськими аеропортами. За оцінкою Міжнародної ради аеропортів (ACI Europe),

					ДР.122.4142.03.ПЗ.Р01	Аркуш
						8
Зм.	Аркуш	№ документа	Підпис	Дата		

Підприємство у 2018 році очолило рейтинг зростання серед великих аеропортів Європи (перше місце в групі європейських аеропортів, які обслуговують від 10 до 25 мільйонів пасажирів).

За результатами 2018 року кредитний рейтинг аеропорту досяг максимального рівня uaAAA, прогноз «стабільний». Підприємство з рейтингом uaAAA характеризується найвищою кредитоспроможністю порівняно з іншими українськими підприємствами або борговими інструментами.

Середньооблікова кількість штатних працівників підприємства у 2018 році 4046 осіб.

Підприємство провадить основну діяльність у трьох сегментах: авіаційні послуги, допоміжні авіаційні послуги та комерційні послуги.

Сегмент авіаційних послуг включає авіаційні послуги, у тому числі використання терміналів та злітно-посадкової смуги, а також забезпечення авіаційної безпеки. Такі послуги, в основному, є об'єктом регулювання.

Сегмент допоміжних авіаційних послуг включає певні послуги з обслуговування пасажирів, наземне обслуговування повітряних суден, забезпечення послуг із заправки паливом, забезпечення харчуванням, а також обслуговування вантажів.

Сегмент комерційних послуг включає надання іншим компаніям площ для діяльності з обслуговування авіаперевізників та пасажирів, для провадження роздрібної торгівлі, для рекламної діяльності, а також надання послуг з паркування автомобілів, готельних послуг, комунальних послуг тощо.

Аеропорт має всі ліцензії та дозволи, необхідні для ведення своєї господарської діяльності.

Підприємство є дійсним членом профільних міжнародних та національних асоціацій: Міжнародна рада аеропортів (Airports Council International; ACI Europe), Українська авіатransпортна Асоціація (УАТА), Торгово-Промислова Палата України, Українська асоціація якості, Організація роботодавців підприємств транспортних послуг, Асоціація платників податків

					ДР.122.4142.03.ПЗ.Р01	Аркуш
						9
Зм.	Аркуш	№ документа	Підпис	Дата		

України тощо, та керується в своїй діяльності стандартами та практиками Міжнародної асоціації повітряного транспорту (International Air Transport Association; IATA), Міжнародної організації цивільної авіації (International Civil Aviation Organization; ICAO).

Підприємство “Аеропорт” є найбільшим та найпотужнішим аеропортом України, що забезпечує більшість авіаційних пасажирських перевезень та значну частину вантажних авіаперевезень.

Попит на послуги аеропорту підтримується вигідним розташуванням на перетині низки міждержавних транспортних шляхів (поєднують Азію з Європою та Америкою), близькістю до столиці, наявністю сучасної інфраструктури та впровадженням стратегії розвитку.

Інфраструктура аеропорту включає в себе 2 злітно-посадкові смуги (довжиною 4 км та 3,6 км), що дозволяють приймати повітряні судна будь якого типу, без обмежень за погодними та світловими умовами, а також 3 термінали (D, B, F). Аеропорт – єдиний в Україні з якого виконуються регулярні трансконтинентальні рейси.

Відповідно до стратегії розвитку Аеропорту, впровадженої з 2015 року, всі міжнародні та внутрішні рейси переведено на обслуговування в термінал D. Це забезпечує зменшення часу на обслуговування пасажирів та суттєве зменшення витрат Підприємства на утримання терміналів B та F, обслуговування яких було тимчасово призупинене. В терміналі F періодично проводилися виставки, презентації та інші заходи, доходи від яких окупали витрати на його утримання. У 2018 році здійснювалась підготовка терміналу F до відновлення роботи як авіаційного пасажирського терміналу з березня 2019 року. Це обумовлено значним зростанням обсягів пасажиропотоку, залученого Підприємством. Крім пасажирських терміналів, функціонує також вантажний термінал [8-10].

					ДР.122.4142.03.ПЗ.Р01	Аркуш
						10
Зм.	Аркуш	№ документа	Підпис	Дата		

Організаційна структура аеропорту представлена на рисунку 1.1.

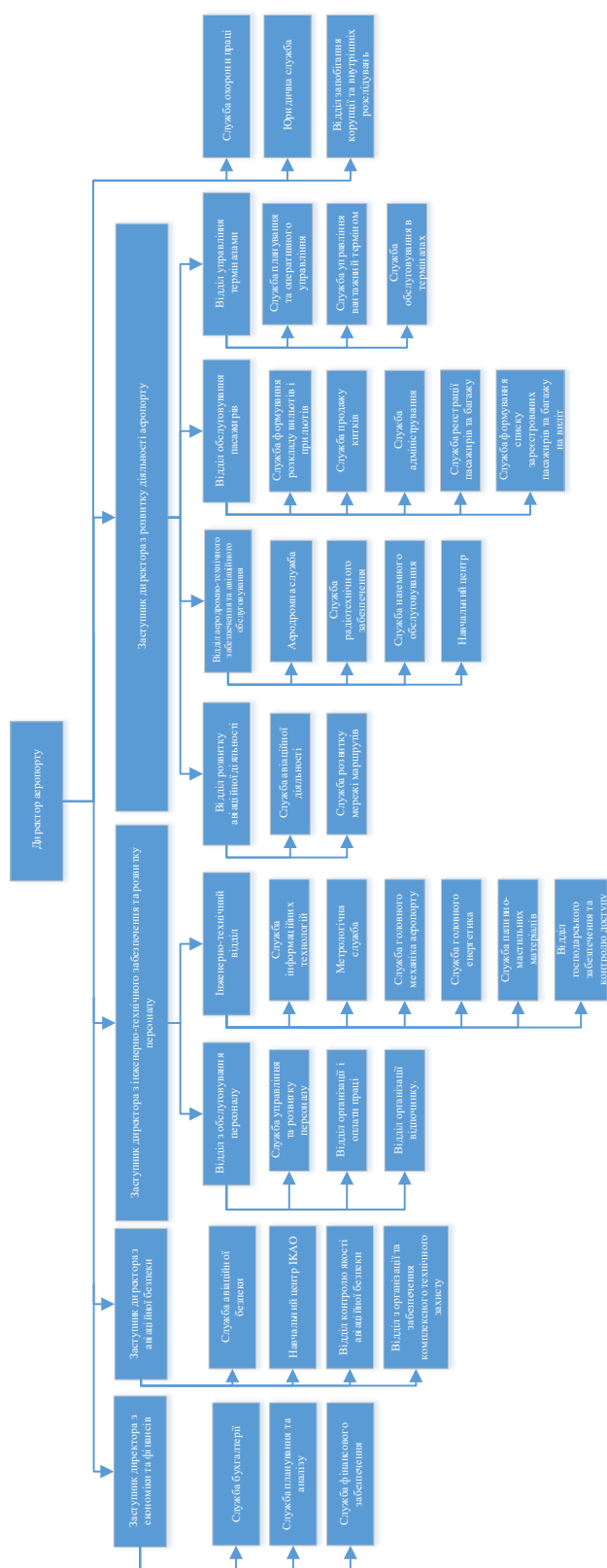


Рисунок 1.1 – Організаційна структура аеропорту

Організаційна структура управління підприємством, має деревоподібну структуру. На чолі неї знаходиться директор аеропорту, керує всіма видами діяльності аеропорту, службою охорони праці, юридичною службою, відділом запобігання корупції та внутрішніх розслідувань, організовує роботу і взаємодію всіх відділів, доручає вирішувати окремі питання своїм заступникам, приймає рішення, затверджує накази. Здійснює координацію та керівництво роботою аеропорту з метою найбільш оптимізувати і підтримувати роботу аеропорту на високому рівні.

Директору аеропорту підпорядковуються:

- заступник директора з економіки та фінансів, який керує службою бухгалтерії, службою планування та аналізу, службою фінансового забезпечення;
- заступник директора з авіаційної безпеки, який керує службою авіаційної безпеки, навчальним центром ІКАО (міжнародна організація цивільної авіації), відділом контролю якості авіаційної безпеки, відділом з організації та забезпечення комплексного технічного захисту;
- заступник директора з інженерно-технічного забезпечення та розвитку персоналу. Він зайнятий роботою з персоналом, стежить за виконанням всіх польотних вимог для екіпажу, організовує навчання пілотів, керує відділом з обслуговування персоналу, який включає службу управління та розвитку персоналу, відділ організації оплати праці, відділ організації відпочинку та керує інженерно-технічним відділом, який включає службу інформаційних технологій, метрологічну службу, відділ господарського забезпечення та контролю доступу, службу головного механіка, головного енергетика аеропорту і службу паливно-мастильних матеріалів;
- заступник директора з розвитку діяльності аеропорту, який керує відділом розвитку авіаційної діяльності, відділом аеродромно-

					ДР.122.4142.03.ПЗ.Р01	Аркуш
						12
Зм.	Аркуш	№ документа	Підпис	Дата		

технічного забезпечення та авіаційного обслуговування, відділом з пасажирського обслуговування, відділом управління терміналами. Відділ розвитку авіаційної діяльності включає службу авіаційної діяльності та службу розвитку мережі маршрутів.

Відділ аеродромно-технічного забезпечення та авіаційного обслуговування включає навчальний центр, службу радіотехнічного забезпечення, службу наземного обслуговування, навчальний центр.

Відділ обслуговування пасажирів включає службу формування розкладу вильотів та прильотів, службу продажу квитків, службу реєстрації багажу та пасажирів, службу адміністрування.

Відділ управління терміналами включає службу планування та оперативного управління, вантажний термінал, службу обслуговування в терміналах.

#### **1.1.2. Аналіз предметної області процесів обробки інформації.**

Так як аеропорт - велика і складна система, буде взято лише частину підприємства, один з основних відділів – відділ обслуговування пасажирів. Діяльність відділу є важливою та одною з основних в аеропорті. Вирішує головні задачі аеропорту: продаж авіаквитків, формування розкладу вильотів та прильотів для потенційних пасажирів.

Для зниження втрати багажу відділ повинен забезпечувати якісну та безпечну реєстрацію пасажирів та багажу. Реєстрація повинна бути оперативна для збільшення швидкості та продуктивності роботи. Обслуговування пасажирів повинно бути якісним, для підняття популярності аеропорту та залучення нових пасажирів, що призводить до збільшення прибутку та конкурентоспроможності аеропорту.

					ДР.122.4142.03.ПЗ.Р01	Аркуш
						13
Зм.	Аркуш	№ документа	Підпис	Дата		

Організаційна структура відділу представлена на рисунку 1.2.

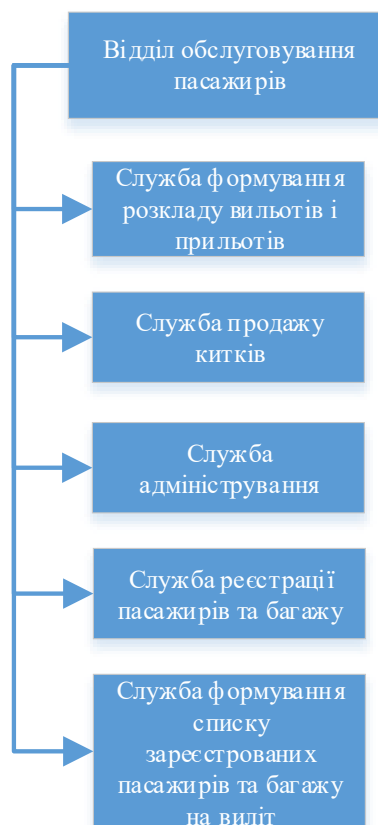


Рисунок 1.2 – Відділ обслуговування пасажирів

Існуюча підсистема в підприємстві характеризується відсутністю сучасних засобів переробки інформації. Виконання багатьох інформаційних операцій співробітником підприємства виконується вручну, також зберігання даних здійснюється в паперових носіях. Служба продажу досить нешвидка та незручна. Підсистема не дає точно фіксувати всі події, які плануються і відбуваються в відділі аеропорту і не надає фахівця всю необхідну інформацію як про плановані прильоти і відправлення повітряних суден, так і статистичну інформацію.

Всі документи оформляються співробітниками не за допомогою автоматизованої системи документообігу, а за допомогою підручних засобів, такі як Word і Excel. Це призводить до величезної кількості помилок, йде в десять разів більше часу на виправлення.

Виходячи з цього, буде розроблено автоматизовану підсистему для відділу аеропорту з обслуговування пасажирів для покращення роботи та підвищення ефективності.

У зв'язку з цим необхідно врахувати всі особливості розробки інформаційної підсистеми для даного типу підприємства, а саме:

- забезпечити безпечний доступ до даних;
- забезпечити розмежування прав доступу;
- реалізувати зрозумілий інтерфейс користувача;
- забезпечити безпечне зберігання даних;
- реалізувати можливість формування та редагування розкладу;
- реалізувати можливість додавання, видалення, редагування інформації;
- реалізувати формування ціни авіаквитків;
- реалізувати якісний продаж авіаквитків;
- реалізувати реєстрацію багажу;
- реалізувати реєстрацію пасажирів;

Користувачами даної ПС є як співробітники аеропорту, так і пасажирі. Пасажирам необхідно забезпечити можливість покупки авіаквитків. Для зручності продажу квитків треба реалізувати сервіс, який дасть можливість клієнту купити авіаквиток, це дозволить зекономити час та знайти нових клієнтів. В більшості випадків, зручніше за все використовувати web-додаток в якості сервісу для продажу квитків, так як є найзручнішим для використання подібних сервісів, потребує лише телефон та доступ до інтернету, котрий являється невід'ємною частиною на сьогоднішній день.

Клієнт має можливість отримати інформацію за розкладом польотів. Диспетчеру необхідно контролювати, формувати розклад вильотів та прильотів. Адміністратору системи необхідно здійснювати адміністрування частин бази даних.

					ДР.122.4142.03.ПЗ.Р01	Аркуш
						15
Зм.	Аркуш	№ документа	Підпис	Дата		



Розробка підсистеми “Обслуговування пасажирів” ІС аеропорту для відділу підприємства дозволить автоматизувати довідкові дані, здійснювати обмін інформацією, також містити сукупність відомостей, якими можуть користуватися всі підрозділи підприємства і користувачі підсистеми, підвищення ефективності роботи співробітників підрозділу.

У зв'язку з цим впровадження нової підсистеми в систему аеропорту є обов'язковим так, як забезпечує безперебійне та оптимальне функціонування при безперервній роботі аеропорту.

## 1.2. Формування вимог до підсистеми

Основними цілями створення ПС:

- оперативна передача даних;
- оперативне отримання даних;
- безпечне зберігання даних;
- підвищення ефективності роботи співробітників організації;
- автоматизація процесів відділу.

Основними задачами ПС є:

- авторизація користувача;
- реєстрація користувача;
- розподіл прав доступу користувачів, доступне користувачам групи «Адміністратор»;
- складання розкладу вильотів і прильотів для пасажирів;
- складання розкладу вильотів і прильотів для диспетчерів;
- формування цін квитків (управління тарифами);
- продаж авіаквитків через сайт;
- продаж авіаквитків через касу;
- реєстрація багажу (майно пасажирів, що транспортується в багажному відділенні);
- реєстрація пасажирів на виліт;

					ДР.122.4142.03.ПЗ.Р01	Аркуш
						16
Зм.	Аркуш	№ документа	Підпис	Дата		

- формування списку багажів на виліт;
- формування списку пасажирів на виліт;
- адміністрування бази даних.

Пошук інформації в підсистемі повинен бути ефективним, за запитом користувача повинно видаватися якомога більше результатів відповідних запиту, Підсистема повинна мати високу швидкість завантаження і обробки інформації для оперативної роботи, мати різні права доступу (клієнт, диспетчер, адміністратор, касирів та інших співробітників аеропорту). Повинні бути засоби для підтримки зворотного зв'язку з користувачем

Підсистема повинна мати базу зберігання інформації і модуль введення-виведення даних, які передбачають web-інтерфейс роботи користувача з ними. Підсистема повинна підтримувати розмежування прав доступу з можливістю формування груп користувачів і присвоєння групі і кожному користувачеві певних повноважень на доступ.

Служба адміністрування підтримує процеси управління базою даних аеропорту і забезпечує її цілісність і працездатність.

Служба адміністрування включає:

- додати літак;
- вилучити літак;
- редагувати інформацію про літак;
- додати пілота;
- вилучити пілота;
- редагувати інформації про пілота;
- додати екіпаж;
- вилучити екіпаж;
- редагувати інформації про екіпаж;
- додати пілота в екіпаж;
- вилучити пілота з екіпажу;
- додати рейс;

					ДР.122.4142.03.ПЗ.Р01	Аркуш
						17
Зм.	Аркуш	№ документа	Підпис	Дата		

- вилучити рейс;
- редагувати рейс;
- додати політ(виліт, приліт);
- вилучити політ;
- редагувати політ;

Модуль вводу-виводу організовує процеси доступу до БД і керує ними. Підсистема адміністрування доступу підтримує розмежування прав доступу.

Надійне функціонування програмного продукту повинно бути забезпечено виконанням організаційно-технічних заходів:

- програмний продукт повинен автоматизувати процеси обробки інформації;
- програмний продукт повинен функціонувати і не приводити до збоїв операційної системи;
- програмний продукт повинен забезпечувати контроль вхідної та вихідної інформації;

Збереження інформації в підсистемі повинно бути забезпечено при виникненні таких аварійних ситуацій:

- програмний збій;
- апаратний збій;
- відключення живлення;

Якщо виникають проблеми з апаратним забезпеченням, включаючи аварійне відключення електроживлення, підсистема повинна автоматично відновлювати свою працездатність після усунення збоїв і коректного перезапуску апаратного забезпечення (по винятком випадків пошкодження робочих носіїв інформації з виконуваним програмним кодом).

Підсистема повинна забезпечувати коректну обробку аварійних ситуацій, викликаних неправильними діями користувачів, невірним форматом або неприпустимими значеннями вхідних даних. У зазначених випадках

					ДР.122.4142.03.ПЗ.Р01	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		18

користувачеві повинні видаватися відповідні аварійні повідомлення, після чого повертатися в робочий стан.

Всі технічні рішення, використані при створенні підсистеми, а також при визначенні вимог до апаратно забезпечення, повинні відповідати чинним нормам і правилам техніки безпеки, пожежної безпеки, а також охорони навколишнього середовища при експлуатації.

Взаємодія користувачів з підсистемою повинно здійснюватися за допомогою візуального графічного інтерфейсу. Введення-виведення даних, прийом керуючих команд і відображення результатів їх виконання повинно виконуватися в інтерактивних режимі, в реальному часі.

Інтерфейс даного програмного продукту повинен бути інтуїтивно зрозумілий будь-якому користувачеві, якому необхідно працювати з інформацією в даній ПС. Також інтерфейс повинен відповідати сучасним ергономічним вимогам і забезпечувати зручний доступ до основних функцій і операцій. Вся інформація повинна бути структурована та надаватися в зрозумілому вигляді.

Інтерфейс повинен бути розрахований на переважне використання маніпулятора типу «миша», тобто управління підсистемою повинно здійснюється за допомогою набору екранних меню, кнопок, значків. Клавіатурний режим введення повинен використовуватися головним чином при заповненні / редагуванні текстових і числових полів екранних форм.

Сторінки повинні бути виконані в єдиному графічному дизайні, з однаковим розташування основних елементів керування та навігації, вживатися подібні графічні значки, кнопки, керуючі елементи. Терміни, що використовуються для позначення типових операцій.

Для відновлення даних у випадках часткової або повної втрати інформації передбачені процедури резервування та архівування. Переносні носії інформації і запасні комплектуючі зберігаються в спеціально відведених для цього місцях.

					ДР.122.4142.03.ПЗ.Р01	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		19

Веб додаток підсистеми включатиме наступні сторінки:

- сторінка реєстрації та входу;
- домашня – коротка інформація про web-додаток, технології створення web додатку;
- розклад польотів – матиме таблицю розкладу вильотів та прильотів та можливість пошуку, сортування;
- сторінка продажу квитків;
- сторінка успішного продажу – матиме інформацію про авіаквиток, ціну квитка.
- сторінка реєстрації багажу;
- сторінка реєстрації пасажирів;
- сторінки призначені для адміністрування бази даних;
- сторінка яка буде надавати список пасажирів та список багажів на виліт;
- сторінка зворотного зв'язку – матиме номер телефона, адресу, електронну пошту для зв'язку в разі необхідності;
- сторінки які повідомляють про різні помилки: квитки на даний рейс закінчилися, квиток вже зайнятий і так далі.

Для управління інформацією, що зберігається в базі даних (БД), а також для її захисту, організації структури БД і забезпечення видачі інформації необхідно використовувати системи управління базами даних (СКБД). База даних повинна зберігатися на сервері. Окрім того необхідно надати можливість одночасної роботи визначеної кількості користувачів із підсистемою в цілому. Варіантами архітектури, що задовольняють поставленим вимогам є клієнт-серверна архітектура.

Програмно-технічні засоби, що розробляється повинні відповідати стандартам мережі Інтернет і підтримувати прийом-передачу даних по протоколу HTTP. Фізичний сервер, на якому розміщуються програмні модулі

					ДР.122.4142.03.ПЗ.Р01	Аркуш
						20
Зм.	Аркуш	№ документа	Підпис	Дата		

автоматизованої інформаційної підсистеми, повинен мати постійне підключення Інтернет по протоколам TCP / IP [1-3].

### 1.3. Аналіз існуючих рішень

На даний момент існує величезна кількість інформаційних систем, що займаються продажами квитків, адмініструванням. Вони можуть являти собою як окремі програми, так і онлайн сервіси.

Розглянуто наступні засоби існуючі на ринку в даний час:

- Інформаційна система Amadeus, розробник Master Price, системні вимоги: ОС Microsoft Windows XP і вище Процесор PIII 1.3 GHz, ОЗУ: 512 Mb.
- Інформаційна система Galilei, розробник Travelport, системні вимоги: ОС: Microsoft Windows XP і вище Процесор: PII - 266, ОЗУ: 128 Mb.
- Інформаційна система Ryanair.com, розробник Ryanair, системні вимоги: вимагає підключення до мережі інтернет.

Amadeus - найпопулярніша система продажів авіаквитків в світі. Є провідним постачальником програмних засобів в області систематизації та дистрибуції інформації. Річний обсяг обслужених клієнтів коливається в районі 500 мільйонів а річний обсяг замовлень перевищує півмільярда [11]. Однак такі показники грають як позитивну роль для системи, тому як запит на рейси здійснюється протягом 40 секунд, що незрівнянно з пошуком того ж рейсу через інтернет-сервіси. також мінусом даної системи є інтерфейс, недружній для ненавченого користувача.

Підключення до системи Amadeus може здійснюватися декількома способами [12]:

- підключення через виділені лінії зв'язку (стаціонарне);
- телефонна версія Dial-Up;
- підключення через інтернет;
- підключення за допомогою мобільного телефону.

					ДР.122.4142.03.ПЗ.Р01	Аркуш
						21
Зм.	Аркуш	№ документа	Підпис	Дата		

Однак абсолютно кожен спосіб вимагає установки і роботи через додаткове ПО системи Amadeus, що ускладнює процес підключення, роблячи його не тільки більш тривалим за часом, але і більш витратним у фінансовому плані.

Galileo - одна з найбільш значимих систем продажів авіаквитків в світі. Забезпечує користувача можливістю бронювати авіап перевезення, супутні туристичні послуги. Також як і Amadeus дана система призначена для роботи на терміналах з навченими користувачами. Підтримує тільки два види підключення: стаціонарне підключення; підключення через інтернет. Стаціонарне підключення здійснюється за допомогою ліній SITA і коштує дорожче підключення через інтернет, але така версія передбачає безкоштовне ПО, що складається з шести програм, які розширюють графічний інтерфейс системи і надають додаткові функції по бронювання. Підключення через інтернет коштує дорожче, проте підтримує багатокористувацький режим [13].

Ryanair.com - сайт авіакомпанії Ryanair, яка є європейським найбільшим бюджетним перевізником. На відміну від інших сайтів, які здійснюють продаж авіаквитків, не використовує жодну з GDS (Global Distribution System – міжнародна комп'ютерна система бронювання). Вся інформація про рейсах, цінах і квитках надається безпосередньо самої авіакомпанією у вигляді БД, а діалог з потенційним покупцем ведеться за допомогою сайту, що оперує до даної БД. Ryanair.com дозволяє здійснювати пошук авіаквитків за різними критеріями. Також при авторизації користувача стає доступний особистий кабінет, що дозволяє переглядати статистику по здійснених перельотам.

Подібна структура системи має як плюси, так і мінуси – дана інформаційна система володіє високою швидкістю і зручним інтерфейсом, зрозумілим пересічному користувачеві але, в силу відсутності підключення до GDS, кількість здійснюваних рейсів, а значить і продаваних квитків обмежена інтересами авіакомпанії [14].

					ДР.122.4142.03.ПЗ.Р01	Аркуш
						22
Зм.	Аркуш	№ документа	Підпис	Дата		

Результати порівняння програмних продуктів представлені в таблиці 1.1

Таблиця 1.1 - Результати порівняння програмних засобів

Назва системи	Переваги	Недоліки
Amadeus	Висока стабільність, незалежна від обсягу даних, багатофункціональний інтерфейс, висока гнучкість адміністрування системи	Низька швидкість роботи, складний для рядового користувача інтерфейс.
Galileo	Кращі способи дистрибуції та управління ресурсами в туристичної індустрії, існує Windows версія системи	Недоступність для рядового користувача. Висока ціна.
Ryanair.com	Простий і зрозумілий інтерфейс, безкоштовне використання	Відсутність підключення до GDS

За результатами проведеного аналізу можна сказати, що на даний момент існує безліч інформаційних систем по продажу квитків та адмініструванню, що володіють великими функціональними засобами. Однак функціонал розглянутих систем не підходять до структури даного підприємства та відділу обслуговування пасажирів. Розроблена підсистема “Обслуговування пасажирів автоматизує основні функції відділу обслуговування пасажирів, вона є сумісною з ІС аеропорту, що обґрунтовує необхідність розробки даної підсистеми.



## **1.4. Розробка концепції підсистеми**

### **1.4.1. Цільове призначення та задача роботи**

Цільовим призначенням роботи є створення підсистеми для підвищення ефективності роботи відділу аеропорту з обслуговування пасажирів, яка дозволить покращити продуктивність співробітників та зробить компанію конкурентоспроможною. Підсистема покращить, автоматизує, збільшить продуктивність роботи відділу з обслуговування пасажирів. Підсистема призначена для клієнтів, працівників аеропорту (диспетчерів, адміністраторів касирів та інших співробітників відділу). Вона повинна забезпечувати зберігання відомостей про літаки, вильоти, прильоти, продажі. Основне завдання це забезпечити якісний продаж (за допомогою сайту або через касу) авіаквитків клієнтам аеропорту, можливість реєстрації багажу, побудови розкладу вильотів та прильотів для потенційних пасажирів і диспетчерської служби.

Використання підсистеми дозволить:

- працювати з великим об'ємом даних;
- зберігати дані протягом тривалого періоду;
- зв'язати кілька компонентів, що мають свої локальні цілі, завдання і прийоми функціонування, в єдину підсистему для роботи з інформацією;
- значно знизити витрати на зберігання і доступ до необхідних даних;
- досить швидко знаходити необхідну інформацію;
- підвищить продуктивність роботи персоналу
- покращить якість обслуговування клієнтів;
- знизити трудомісткість і напруженість праці персоналу, мінімізує помилки в його діях.

Підсистема забезпечить безперервне функціонування частини аеропорту по обслуговуванні пасажирів, за допомогою бази даних допоможе надійно і структуровано зберігати інформації, забезпечить автоматизацію процесів

					ДР.122.4142.03.ПЗ.Р01	Аркуш
						24
Зм.	Аркуш	№ документа	Підпис	Дата		

обробки інформації аеропорту, дасть можливість купити авіаквитки за декілька хвилин, сидячи вдома за допомогою сайту або на касі в аеропорту.

#### **1.4.2. Розробка функціональної моделі підсистеми**

Метою побудови функціональних моделей зазвичай є виявлення найбільш слабких і уразливих місць діяльності організації, аналіз переваг нових бізнес-процесів і ступеня зміни існуючої структури організації бізнесу.

Функціональну модель підсистеми можна представити у вигляді блок-схеми, на якій кожен елемент підсистеми представляється у вигляді блоку, а зв'язки і їх напрямки вказуються стрілками [4].

На основі проведеного аналізу структури відділу аеропорту з обслуговування пасажирів, сформованих вимог до підсистеми, які були представлені в даній роботі та з метою визначення інформаційних потоків процесів, які повинні відбуватися в підсистемі розроблено функціональну модель підсистеми.

Модель підсистеми “Обслуговування пасажирів” ІС аеропорту представлена у вигляді блок-схеми на рисунку 1.3

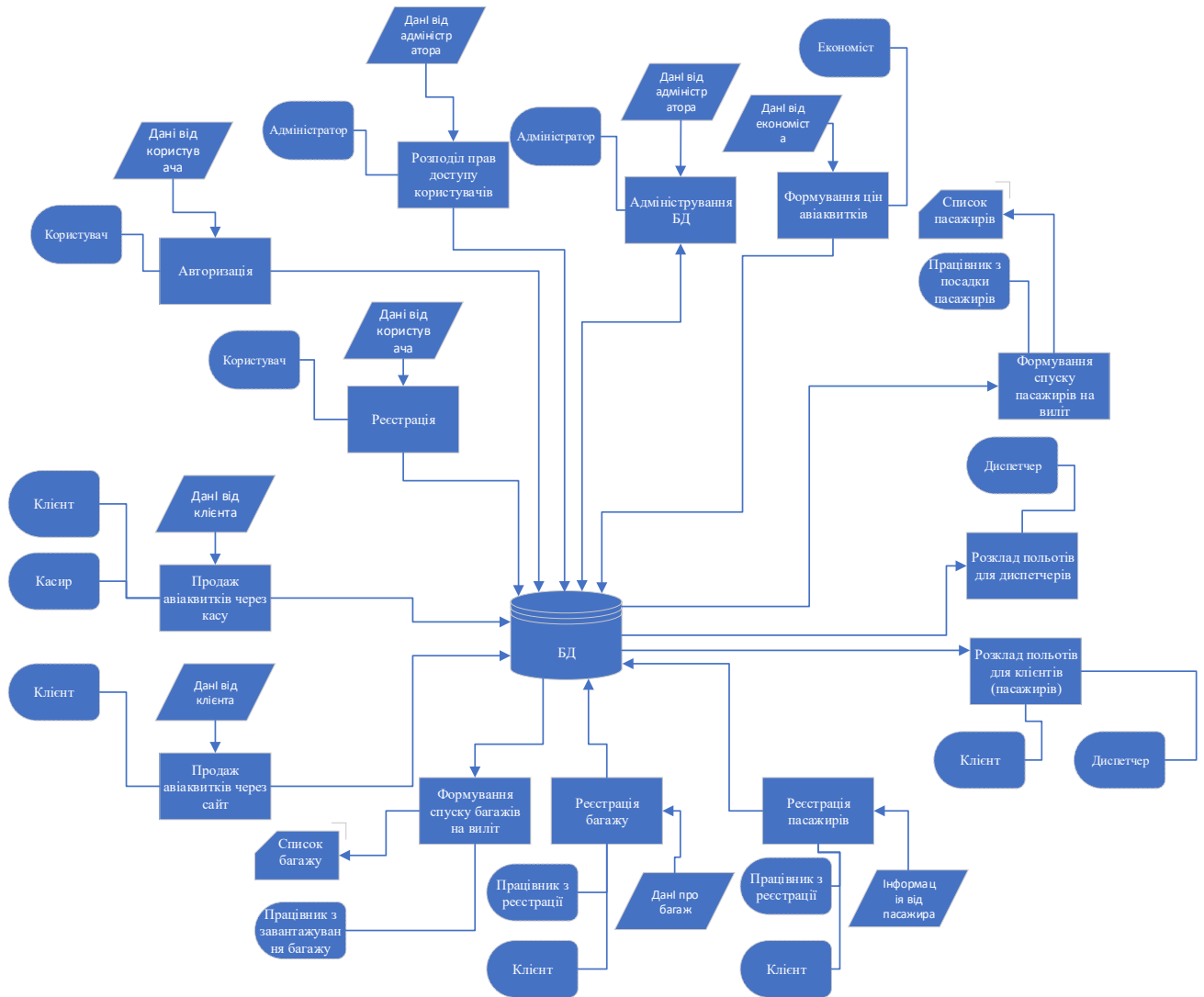


Рисунок 1.3 – Модель підсистеми “Обслуговування пасажирів” ІС аеропорту

Опис служб підсистеми:

- реєстрація – надає користувачу можливість зареєструватися в підсистемі
- авторизація – надає можливість користувачу увійти в підсистему (авторизуватися)
- розподіл прав доступу користувачів – надає можливість адміністратору керувати, розподіляти права доступу до підсистеми

- продаж авіаквитків через сайт – надає можливість клієнту за допомогою web-додатку за короткий час придбати квиток, надавши персональні дані, на необхідний рейс та вибрану дату вильоту.
- продаж авіаквитків через касу – надає можливість клієнту за допомогою касира придбати авіаквиток. Клієнт надає всю необхідну інформацію касиру. Касир в свою чергу за допомогою підсистеми (web – додатку) здійснює продаж.
- формування ціни на авіаквитки – надає можливість змінювати тариф та суму аеропортових зборів в залежності від вильоту.
- реєстрація багажу – надає можливість здати свій багаж в аеропорту. Багаж забирають співробітники аеропорту, зважують, у разі великої ваги ( більше 20 кг ), багаж не приймається. Якщо вага в допустимій нормі інформація про багаж заноситься в базу даних і пасажиру надається номер багажу. Для пасажирів економ-класу надається одне безкоштовне місце реєстрованого багажу до 20 кг. Для пасажирів першого і бізнес-класу передбачається два багажних місця до 20 кг.
- реєстрація пасажирів – надає можливість зареєструвати пасажирів на виліт. Для реєстрації пасажир надає номер квитка та паспортні дані. В результаті успішної реєстрації пасажир отримує посадковий талон із зазначенням місця.
- розклад польотів для диспетчерів – надає можливість диспетчерам аеропорту використовувати, оновлювати, сортувати розклад вильотів та прильотів. В розкладі диспетчерів надається повна інформація про стан польотів
- розклад польотів для пасажирів – надає можливість пасажирам переглядати плановий розклад вильотів та прильотів літаків, використовуючи web – додаток, Розклад для пасажирів формується диспетчерами кожні два тижні.

					ДР.122.4142.03.ПЗ.Р01	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		27

- формування списку пасажирів на виліт - формується список пасажирів на певний виліт, який надається (в електронному або друкованому вигляді) співробітникам, які займаються процесом посадки і прийому пасажирів.
- формування списку багажів на виліт – формується список багажу на певний виліт, який надається (в електронному або друкованому вигляді) співробітникам, які займається завантажуванням і розвантажуванням багажів в вантажні відділи літаків.
- адміністрування – надає можливість адміністраторам виконувати адміністрування бази даних: додавання, вилучення, редагування, перегляд інформації.

#### **1.4.3. Формування вхідної, довідкової та вихідної інформації бази даних підсистеми**

Відповідно до кожного процесу, які описані в пункті 1.3.2. потрібно виділити вхідну, довідкову та вихідну інформацію бази даних підсистеми:

- інформація від користувача для реєстрації: електронна пошта, логін, пароль;
- інформація від користувача для авторизації: електронна пошта або логін и пароль;
- вхідна інформація для розподілу прав доступу: роль користувача;
- інформація від клієнта для покупки авіаквитка через сайт або касу: прізвище, ім'я, вік, номер телефону, електронна пошта, пункт рейс, дата вильоту, серія паспорта, номер паспорта, клас (економ, бізнес, перший). Вихідною інформацією є ціна квитка, номер квитка (ідентифікатор квитка).
- ціна на квиток складається з тарифу та аеропортових зборів. Повернення квитків неможливе. Після покупки авіаквитка, він буде відправлений на вказану електронну пошту. Тариф та аеропортові

					ДР.122.4142.03.ПЗ.Р01	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		<b>28</b>

збори формуються на кожний політ. Вхідна інформація для створення тарифів: назва тарифу, грошова сума. Вихідною інформацією є ідентифікатор нового тарифу. Вхідна інформація для створення аеропортових зборів: назва аеропортових зборів, грошова сума. Вихідною інформацією є ідентифікатор нового тарифу.

- для реєстрація пасажирів пасажири треба надати дані: номер квитка (номер продажу), ім'я, прізвище, серію паспорта, номер паспорта. Персональні дані надаються для порівняння з даними в базі підсистеми. Вихідною інформацією є номер посадкового місця.
- для реєстрації багажу пасажиру треба надати дані: номер квитка (номер продажу), ім'я, прізвище, серію паспорта, номер паспорта. Вихідною інформацією є номер багажу, вага багажу.
- Вихідною інформацією в розкладі польотів для пасажирів є назва рейс, назва екіпажу, день вильоту, дата вильоту, дата прильоту, тривалість рейсу, тип рейсу (прямий або непрямий).
- Вихідною інформацією в розкладі польотів для пасажирів є назва рейсу, назва екіпажу, бренд літака (назва літака), день вильоту, дата вильоту, дата прильоту, тривалість рейсу, тип рейсу (прямий або непрямий).
- Вихідною інформацією в розкладі польотів для диспетчерів є назва рейсу, пункт відправлення, пункт призначення, назва екіпажу, інформація про пілотів екіпажу (ім'я, прізвище, посада, номер телефона, дата народження), назва літака, бортовий номер літака, швидкість літака, паливо літака, кількість посадкових місць, день вильоту, дата вильоту, дата прильоту, тривалість рейсу, тип рейсу (прямий або непрямий), кількість куплених квитків, кількість зареєстрованого багажу, кількість зареєстрованих пасажирів.
- Вхідною інформацією для формування списку пасажирів на виліт є ідентифікатор вильоту. Вихідною інформацією є список пасажирів,

					ДР.122.4142.03.ПЗ.Р01	Аркуш
						29
Зм.	Аркуш	№ документа	Підпис	Дата		

який містить інформацію ім'я, прізвище, серія паспорта, номер паспорта, електронна пошта, номер телефону кожного зареєстрованого на виліт пасажирів.

- Вхідною інформацією для формування списку багажу на виліт є ідентифікатор вильоту. Вихідною інформацією є номер багажу, номер продажу квитка (номер квитка), ім'я, прізвище, номер паспорта, серія паспорта пасажирів, якому належить багаж.
- Вхідною інформація для додавання рейсу в підсистему є назва рейсу, авіакомпанія, час вильоту, час прибуття, пункту відправлення, пункту призначення, тривалість рейсу, тип рейсу. Вихідною інформацією є ідентифікатор нового рейсу. Пункти призначення та відправлення містять інформацію: назва міста, назва країни, назва пункту.
- Вхідною інформація для додавання літака в підсистему є марка літака, максимальна швидкість, бортовий номер, кількість посадкових місць, паливо. Вихідною інформацією є ідентифікатор нового літака.
- Вхідною інформація для додавання пілота в підсистему є номер пілота, посада, екіпаж, прізвище, ім'я, дата народження, адреса, телефон. Посада включає: назва посади, оклад, премія. Вихідною інформацією є ідентифікатор нового пілота.
- Вхідною інформація для додавання екіпажу в підсистему назва екіпажу, стаж екіпажу. Довідковою інформацією є кількість пілотів в екіпажі. Вихідною інформацією є ідентифікатор нового літака.
- Вхідна інформація для додавання пілота в екіпаж є номер пілота.
- Вхідна інформація для вилучення пілота з екіпажу є номер пілота.
- Вхідна інформація для додавання польоту в підсистему є номер польоту, рейс, екіпаж, літак, дата вильоту, кількість зареєстрованих пасажирів, кількість проданих квитків, кількість зареєстрованого багажу, тариф, аеропортовий збір. Вихідною інформацією є ідентифікатор нового польоту.

					ДР.122.4142.03.ПЗ.Р01	Аркуш
						30
Зм.	Аркуш	№ документа	Підпис	Дата		

Після проведення обстеження предметної області була виділена наступна вхідна, довідкова та вихідна інформація, яка буде зберігатися в базі даних підсистеми:

- Рейси (номер рейсу, назва рейсу, авіакомпанія, час вильоту, час прибуття, пункту відправлення, пункту призначення, тривалість рейсу);
- Літаки (номер літака, марка літака, максимальна швидкість, бортовий номер, кількість посадкових місць, паливо);
- Пілоти (номер пілота, посада, екіпаж, прізвище, ім'я, дата народження, адреса, телефон);
- Екіпажі (номер екіпажу, назва екіпажу, кількість пілотів в екіпажі, стаж екіпажу);
- Посади (номер посади, назва посади, оклад, премія);
- Пункти (Аеропорти) (номер пункту, назва міста, назва країни, назва пункту);
- Авіакомпанії (номер авіакомпанії, назва авіакомпанії);
- Польоти (номер польоту, рейс, екіпаж, літак, дата, кількість зареєстрованих пасажирів, кількість проданих квитків, кількість зареєстрованого багажу, тариф, аеропортовий збір);
- Продаж квитків (номер, виліт, прізвище, ім'я, вік, номер телефону, електронна пошта, серія паспорта, номер паспорта, номер місця, проходження реєстрації (true or false), клас, ціна);
- Багажі (номер багажу, квиток (номер продажі квитка), вага багажу).
- Тарифи (номер тарифу, назва тарифу, сума);
- Аеропортові збори (номер зборів, назва зборів, сума);
- Користувач (логін, пароль, email);
- Роль доступу (назва ролі);

					ДР.122.4142.03.ПЗ.Р01	Аркуш
						31
Зм.	Аркуш	№ документа	Підпис	Дата		



#### 1.4.4. Вибір ОС та програмного інструментарію для реалізації підсистеми

Операційна система – це базовий комплекс програм, що виконує керування апаратною складовою комп'ютера або віртуальної машини; забезпечує керування обчислювальним процесом і організовує взаємодію з користувачем.

Вибір операційної системи для реалізації системи є важливим етапом розробки, яким не можна нехтувати, адже від цього залежить які інструменти будуть доступні, які мінімальні системні вимоги, якого рівня безпеки можна досягти, яка буде стабільність роботи, який рівень підтримки, доступність самої ОС та звичайно зручність використання.

Вибір здійснювався серед трьох основних операційних систем: Windows, Linux, Mac OS. Коротко розглянемо кожен з 3 операційних систем, їхні основні переваги та недоліки.

Windows – Сімейство комерційних операційних систем корпорації Microsoft орієнтовані на застосування графічного інтерфейсу при управлінні.

Плюси:

- Сумісність. Практично будь-яка програма або працює, або має аналоги під Windows. Драйвери є для будь-яких пристроїв.
- Підтримка. Компанія Microsoft пропонує відмінну підтримку, тому програми легко встановлювати. А ось, наприклад, з Linux подібної підтримки не дістати.
- Доступність. Багато корисних функцій. Інтерфейс досить дружній, зрозумілий на інтуїтивному рівні, в ньому зможе розібратися навіть новачок.

Мінуси:

- Віруси. Більшість вірусів націлені на Windows.

- Швидкість. Windows потребує багато ресурсів. У ньому маса зайвого для звичайного користувача і не професіонал не зможе розібратися, що зайве, а що ні.
- Ціна. Після кожного великого релізу доведеться купувати ОС заново. Винятком стала тільки 10 версія з тимчасовим безкоштовним оновленням.

Mac OS - сімейство операційних систем виробництва корпорації Apple.

Плюси:

- Віруси. У цій ОС їх практично немає.
- Надійність. Система була написана під конкретні «залізні» набори (конфігурації комп'ютерів). Виходячи з цього, функціонує система так, як повинна і рідко ламається.
- Оптимізація ПО. Додатків існує не багато, але вони пишуться для ідеальної сумісності з системою і в підсумку Mac OS працює стабільно набагато довше, ніж інші, при цьому ресурси витрачаються раціонально.
- Дизайн. Зовні багато пристроїв Apple вигаршно виділяються серед конкурентів.

Мінуси:

- Ціна. Mac є дорогим в співвідношенні ціна-залізо. Однак це компенсується відмінною якістю.
- Внутрішня начинка. Apple використовує власні стандарти і нав'язує їх користувачам. Як результат - змінюючи пристрій, користувач повинен звикати до особливостей використання.
- Фізичний комп'ютер. Mac OS розрахована на конкретні конфігурації, і тому вкрай не рекомендується встановлювати її на комп'ютер не з сімейства Apple.
- Сумісність. Кількість програм, написаних під Mac, сильно поступається кількості програм під Windows.

					ДР.122.4142.03.ПЗ.Р01	Аркуш
						33
Зм.	Аркуш	№ документа	Підпис	Дата		

Linux - Для цієї системи досить складно дати однозначну відповідь, оскільки існує багато дистрибутивів Linux, і вони, часом, сильно відрізняються. Якщо в загальних рисах описувати характеристики, тоді можна виділити наступне:

Плюси:

- Малі технічні вимоги. Цій системі досить одного ядра процесора і від 256 до 512 МБ оперативної пам'яті для того, щоб працювати.
- Ціна. Багато дистрибутивів безкоштовні. Їх можна встановити і внести свою лепту в розробку. Повністю модифікувати систему під себе.
- Різноманітність. Велика кількість версій. Linux - тільки ядро і щоб його використовувати, необхідно багато додаткового софту, тому існує близько сотні різних дистрибутивів. Найпопулярнішими є Ubuntu, Fedora, CentOS і Mint.
- Просте застосування. Деякі дистрибутиви досить прості у використанні, є спеціальні з графічним інтерфейсом, наприклад Ubuntu чи Mint.
- Віруси. Їх також, як і в Mac, практично немає.

Мінуси:

- Сумісність. Багато ПЗ з відкритим кодом, де можна вносити свої зміни. Це вимагає багато часу і зусиль. Бувають моменти, коли драйверів для ваших пристроїв не існує, і їх потрібно або писати самому з нуля, або використовувати щось підходяще під всі типи даного пристрою.

Функціональність пристроїв від цього страждає.

Після порівняння операційних систем для реалізації підсистеми було обрано ОС Windows 10.

В якості програмного інструментарію було обрано набір вільного програмного забезпечення .NET який входять веб-сервер IIS Express, СУБД MSSQL Server, та середу виконання CLR. Такий набір програм

					ДР.122.4142.03.ПЗ.Р01	Аркуш
						34
Зм.	Аркуш	№ документа	Підпис	Дата		

було обрано через свою гнучкість, продуктивність та низьку вартості (всі її складові є відкритими і можуть бути безкоштовно завантажені з Інтернету).

Microsoft SQL Server - система керування базами даних (СКБД), розроблена корпорацією Microsoft.

Microsoft SQL Server як мову запитів використовує версію SQL, що отримала назву Transact-SQL (скорочено T-SQL), яка є реалізацією SQL-92 (стандарт ISO для SQL) з багатьма розширеннями. T-SQL дозволяє використовувати додатковий синтаксис процедур, що зберігаються і забезпечує підтримку транзакцій (взаємодія бази даних з керуючим застосунком). Microsoft SQL Server та Sybase ASE для взаємодії з мережею використовують протокол рівня застосунка під назвою Tabular Data Stream (TDS, протокол передачі табличних даних).

Microsoft SQL Server також підтримує Open Database Connectivity (ODBC)—інтерфейс взаємодії застосунків з СУБД. Microsoft SQL Server надає можливість підключення користувачів через веб-сервер-сервіси, що використовують протокол SOAP. Це дозволяє клієнтським програмам, не призначеним для Windows, кроссплатформенно з'єднуватися з SQL Server.

Додаток буде реалізовано на платформі Microsoft .NET - представляє потужну платформу для створення додатків. Основні риси платформи [15]:

- Підтримка декількох мов. Основою платформи є загальномовне середовище виконання Common Language Runtime (CLR), завдяки чому .NET підтримує кілька мов: поряд з C # це також VB.NET, C++, F#, а також різні діалекти інших мов, прив'язані до .NET, наприклад, Delphi. NET. При компіляції код на будь-якому з цих мов компілюється в збірку спільною мовою CIL (Common Intermediate Language) - свого роду асемблер платформи .NET. Тому можливо зробити окремі модулі однієї програми на окремих мовах.
- Кроссплатформеність. .NET є переносимою платформою. Наприклад, остання версія платформи на даний момент .NET Core підтримується

на більшості сучасних ОС Windows, MacOS, Linux. Використовуючи різні технології на платформі .NET, можна розробляти програми на мові C# для самих різних платформ - Windows, MacOS, Linux, Android, iOS, Tizen.

- Потужна бібліотека класів. .NET представляє єдину для всіх підтримуваних мов бібліотеку класів.
- Різноманітність технологій. Загальномовне середовище виконання CLR і базова бібліотека класів є основою для цілого стека технологій. Наприклад, для роботи з базами даних в стеку технологій призначена технологія ADO.NET і Entity Framework Core. Для побудови графічних додатків з багатим насиченим інтерфейсом - технологія WPF і UWP, для створення більш простих графічних додатків - Windows Forms. Для розробки мобільних додатків - Xamarin. Для створення веб-сайтів - ASP.NET.

В якості IDE для реалізації підсистеми обрано середовище програмування Microsoft Visual Studio 2019 [15].

Дане середовище розробки має наступні позитивні характеристики:

- Підтримує програмну платформу .NET Core та .NET Framework, що дозволяє виконувати програми написані на різних мовах програмування, що забезпечує гнучкість при розробці.
- Включає локальний сервер ASP.NET Development Server для перевірки працездатності web-додатку без встановлення стороннього ПО.
- Зручний інтерфейс, підтримка фірми-виробника Microsoft і широкий асортимент навчальних статей та прикладів по розробці програмного забезпечення.
- Безплатна версія Microsoft Visual Studio Community, яка включає всі необхідні компоненти для розробки програмного забезпечення.

В якості мови програмування обрано мову програмування C# - об'єктно-орієнтована мова програмування. C# відноситься до сім'ї мов з C-подібним

					ДР.122.4142.03.ПЗ.Р01	Аркуш
						36
Зм.	Аркуш	№ документа	Підпис	Дата		

синтаксисом, з них його синтаксис найбільш близький до C++ і Java. Мова має статичну типізацію, підтримує поліморфізм, перевантаження операторів (в тому числі операторів явного і неявного приведення типу), делегати, атрибути, події, властивості, узагальнені типи і методи, ітератори, анонімні функції з підтримкою замикань, LINQ, виключення, коментарі у форматі XML.

Переїнявши багато від своїх попередників - мов C++, Delphi, і Java - C#, спираючись на практику їх використання, виключає деякі моделі, що зарекомендували себе як проблематичні при розробці програмних систем.

Для розробки дизайну і верстки сторінок обрано технології HTML, CSS, Bootstrap, JavaScript. Ці технології є найбільш поширеними при розробці верстки та дизайну інтернет сторінок, а також були розглянуті в рамках навчального курсу.

#### 1.4.5. Визначення засобів обчислювальної техніки

На основі рішень з програмного забезпечення представимо вимоги до технічного забезпечення. Оскільки в ролі ОС для реалізації системи було обрано Windows 10, то мінімальні системні вимоги є наступними:

- процесор або система на кристалі (SoC) з частотою 1 ГГц чи вищою;
- Оперативна пам'ять: 1 ГБ (для 32-розрядної версії) або 2 ГБ (для 64-розрядної версії);
- Простір на жорсткому диску: 16 ГБ (для 32-розрядної версії) або 32 ГБ (для 64-розрядної версії);
- Відеокарта: DirectX 9 або новішої версії;

Рекомендовані системні вимоги:

- Процесор: (Intel i3 / i5 / i7 / i9-7x), Core M3-7xxx, Xeon E3-xxxx и Xeon E5-xxxx, AMD (А-серії Ах-9xxx, Е-Серія)
- Оперативна пам'ять: 4 ГБ (для 32-розрядної версії) або 8 ГБ для (для 64-розрядної версії)
- SSD :не менше 128 ГБ для 64-розрядної и 32-розрядної

- Відеокарта: DirectX 9 або новішої версії;

Оскільки при реалізації системи будуть використовуватися бібліотеки, які необхідно завантажити в програму з віддалених репозиторіїв, а також необхідний доступ до документації по програмних інструментах, то обов'язковою вимогою до технічного забезпечення є наявність інтернет з'єднання.

### 1.5. Постановка задачі

На підставі аналізу, вимог і обмежень до інформаційної підсистеми, концепції, а також вибраних ОС та програмного інструментарію для реалізації підсистеми, сформульовано опис постановки задачі.

Необхідно розробити підсистему для управління відділом аеропорту з обслуговування пасажирів, що буде задовольняти перерахованим вище функціональним вимогам до підсистеми. Необхідно забезпечити програмний контроль вхідної та вихідної, надійність збереження інформації та безпечність доступу до зберігаємої інформації в підсистемі. Забезпечити розмежування прав доступу до та можливість одночасного обслуговування багатьох користувачів.

Підсистема повинна мати модулі:

*Модуль реєстрації та авторизації* для реєстрації та входу в підсистему.

*Модуль управління правами доступу* для розподілення прав доступу користувачів.

*Модуль продажу квитків* використовується для купівлі квитка, дозволяє користувача (пасажиру) купити квиток на необхідний рейс та дату.

*Модуль формування розкладу для пасажирів* надає змогу пасажирам переглядати розклад польотів ( вильотів та прильотів ), та надає можливість сортування та пошуку.

*Модуль формування розкладу для диспетчерів* надає можливість переглядати, сортувати, редагувати повний розклад польотів для забезпечення безпечного відправлення та прийому літаків.

*Модуль адміністрування польотів* використовується для введення даних про польотів у підсистему, дозволяє змінювати характеристики польотів у підсистемі, переглядати та видаляти їх.

*Модуль адміністрування літаків* використовується для введення даних про літаків у підсистему, дозволяє змінювати характеристики літаків у підсистемі, переглядати та видаляти їх.

*Модуль адміністрування екіпажів* використовується для введення даних про екіпажи у підсистему, дозволяє змінювати характеристики екіпажів у підсистемі, переглядати та видаляти їх. Надає можливість додавати пілотів в екіпаж та вилучати їх з екіпажу.

*Модуль адміністрування пілотів* використовується для введення даних про пілотів та посад пілотів у підсистему, дозволяє змінювати характеристики пілотів та посад у підсистемі, переглядати та видаляти їх.

*Модуль адміністрування рейсів* використовується для введення даних про рейси, пункти призначення, відправлення у підсистему, дозволяє змінювати характеристики рейсів та пунктів(аеропортів) у підсистемі, переглядати та видаляти їх.

*Модуль реєстрація багажу* використовується для реєстрації багажу на виліт. Надає можливість ведення даних про багажі в підсистему, дозволяє змінювати характеристики багажу та видаляти їх.

*Модуль реєстрації пасажирів* використовується для реєстрації пасажирів на виліт. Пасажир надає квиток, та персональні дані для порівняння з даними в підсистемі (в продажі квитка).

*Модуль генерації списку пасажирів на виліт* використовується для генерації списку зареєстрованих пасажирів на виліт. Надає можливість переглядати списки пасажирів на різні вильоти.

					ДР.122.4142.03.ПЗ.Р01	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		39



*Модуль генерації списку багажу на виліт* використовується для генерації списку зареєстрованого багажу на виліт. Надає можливість переглядати списки на різні вильоти.

*Модуль формування ціни* використовується для визначення аеропортових зборів та тарифів на виліт. Надає можливість додавання тарифів та аеропортових зборів, дозволяє змінювати грошову суму тарифів та аеропортових зборів, переглядати та видаляти їх.

Програмне забезпечення повинно бути реалізована на основі концепції Model-View-Controller (MVC) - схема поділу даних програми, призначеного для користувача інтерфейсу і керуючої логіки на три окремих компоненти: модель, вигляд і контролер - таким чином модифікація кожного компонента може здійснюватися незалежно.

Концепція MVC розділяє дані, вигляди та обробку дій користувача на компоненти [16]:

Модель (Model) - надає собою об'єктну модель якоїсь предметної області, включає в себе дані і методи роботи з цими даними: запити до бази даних, перевірка на коректність. Реагує на запити з контролера, повертаючи дані і / або змінюючи свій стан. При цьому модель не містить в собі інформації про способи візуалізації даних або форматах їх подання, а також не взаємодіє з користувачем безпосередньо.

Вигляд (View) - відповідає за відображення інформації (візуалізацію). Вигляд не обробляє ведені дані користувача. Одні і ті ж дані можуть представлятися різними способами і в різних форматах. Наприклад, колекцію об'єктів за допомогою різних уявлень можна уявити на рівні користувача інтерфейсу як в табличному вигляді, так і списком; на рівні API можна експортувати дані як в JSON, так в XML.

Контролер (Controller) - забезпечує зв'язок між користувачем і системою, використовує модель і уявлення для реалізації необхідної реакції на дії користувача. Як правило, на рівні контролера здійснюється фільтрація

					ДР.122.4142.03.ПЗ.Р01	Аркуш
						40
Зм.	Аркуш	№ документа	Підпис	Дата		

отриманих даних і авторизація - перевіряються права користувача на виконання дій або отримання інформації.

Клієнтський рівень повинен бути побудований за допомогою мови програмування JavaScript та технологій HTML, CSS, Bootstrap. Серверний рівень за допомогою мови програмування C# та фреймворків ASP.NET, Entity Framework, як базу даних було обрано MSSQL. Інформаційна підсистема повинна розроблятися за допомогою засобів Microsoft Visual Studio 2019.

## 1.6. Висновки

В першому розділі було проведено аналіз присутніх проблем в підсистемі відділу з обслуговування пасажирів та поставлено вимоги до нової підсистеми, яка б була позбавлена виявлених проблем. Розроблено концепцію підсистеми. Описано об'єкт автоматизації та його організаційну структуру. Розроблено функціональні, програмні та технічні вимоги до підсистеми, що дозволяють якісно поліпшити роботу підприємства. Визначено ціль та задачі створення підсистеми. Розроблено функціональну модель блок-схему підсистеми. Та детально описано всі функції даної підсистеми. Сформовано вхідну, довідкову та вихідну інформацію бази даних підсистеми, яка допоможе правильно спроектувати базу даних. Визначено ОС та програмний інструментарій для розробки підсистеми, що дозволить розробити надійний, швидкий та з гарним інтерфейсом додаток для підсистеми.

Затрати на розробку підсистеми окупляться за рахунок покращення якості обслуговування пасажирів, служби продажу квитків, що збільшить надходження нових клієнтів, конкурентоспроможність, популярність, прибуток підприємства. Свою собівартість підсистема окупить за один рік і буде приносити прибуток, подальша розробка проекту є доцільна, проект є економічно вигідним.

					ДР.122.4142.03.ПЗ.Р01	Аркуш
						41
Зм.	Аркуш	№ документа	Підпис	Дата		

					ДР.122.4142.03.ПЗ.Р02			
Зм.	Аркаш	№ документа	Підпис	Дата				
					Розробка технічного проекту підсистеми “Обслуговування пасажирів” ІС аеропорту	Літ.	Аркуш	Аркушів
Студент	Борис А. В.						42	32
Керівник	Маршак О. І.					НУК		
Зав. каф.	Михелєв І. Л.							

## **2. РОЗРОБКА ТЕХНІЧНОГО ПРОЕКТУ ПІДСИСТЕМИ “ОБСЛУГОВУВАННЯ ПАСАЖИРІВ” ІС АЕРОПОРТУ**

### **2.1. Загальносистемні рішення підсистеми**

#### **2.1.1. Організаційна структура**

Підсистема призначена для частини підприємства, яка займається обслуговуванням пасажирів та розробкою розкладів прийомів та відправлень літаків міжнародних та внутрішніх рейсів.

За функціонуванням відділу обслуговування пасажирів стежить начальник, який підпорядковується заступнику директора з розвитку діяльності аеропорту. Він керує: завідувачем квитковими касами, який організовує роботу квиткових кас та касирів, завідувачим служби реєстрації багажу та пасажирів, який організовує реєстрацію багажу та пасажирів на виліт, адміністраторами підсистеми в обов'язок, яких входить адміністрування бази даних, економістами (формують тарифи та аеропортові збори на рейси), завідувачим по прийому та відправленню літаків в обов'язок якого входить контроль за безпечним транспортуванням багажу, надання списку пасажирів та багажів на виліт. Списки надаються працівники відділу, які займаються процесом посадки і прийому пасажирів, та працівники, які займаються завантажуванням і розвантажуванням багажів в вантажні відділи літаків. Це підвищує структурованість, безпеку та автоматизує процес посадки та висадки пасажирів, зменшує втрату багажу.

Також начальник відділу керує завідувачим диспетчерської служби, який організовує роботу диспетчерів. Диспетчерська служба займаються побудовою розкладів вильотів, прильотів для пасажирів, та розробкою повного розкладу для самих себе, для повного контролю за польотами аеропорту.

Розроблена інформаційна підсистема автоматизує функціонування відділу з обслуговування пасажирів, підвищить ефективність роботи співробітників підрозділу.

					ДР.122.4142.03.ПЗ.Р02	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		<b>43</b>

Склад посадових осіб (підрозділів) відділу, які забезпечують функціонування підсистеми або під час прийняття рішення використовують інформацію, отриману від її експлуатації, представлено на рисунку 2.1.



Рисунок 2.1 – склад осіб відділу обслуговування пасажирів

### 2.1.2. Функціональна структура

Функціональна структура є однією з найбільш поширених організаційних структур. В рамках цієї структури організація групує співробітників відповідно до спеціалізованим або схожим набором ролей або завдань. Функціональна структура включає підсистеми (комплекси задач, завдання, модулі); автоматизовані функції і сукупності дій (операцій), що виконуються при реалізації автоматизованих функцій тільки технічними засобами (автоматичні) або тільки людиною. Призводить до підвищення ефективності і підвищенню продуктивності. Завдяки своєму досвіду працівники зі спеціальними навичками можуть виконувати завдання швидко, ефективно і з більшою впевненістю, тим самим зменшуючи кількість помилок, пов'язаних з роботою. Основна мета функціональних структур в організаціях - об'єднати всі людські та інформаційні ресурси для досягнення цілей організації.

Підсистема в процесі свого функціонування виконує певні функції, які представлені в схемі функціональної структури на рисунку 2.2.



Рисунок 2.2 – Функціональна структура підсистеми

Опис функцій підсистеми:

- F1.1) Надає змогу користувачу підсистеми (економісту, адміністратору) додавати тарифи. Вхідні дані: назва тарифу, грошова сума.

Процеси: реалізувати інтерфейс користувача для додавання тарифів.

- F1.2) Надає змогу користувачу підсистеми (економісту, адміністратору) вилучати тарифи.

Процеси: реалізувати інтерфейс користувача для вилучення тарифів.

- F1.3) Надає змогу користувачу підсистеми (економісту, адміністратору) редагувати тарифи.

Процеси: реалізувати інтерфейс користувача для редагування тарифів.

- F1.4) Надає змогу користувачу підсистеми (економісту, адміністратору) додавати аеропортові збори. Вхідні дані: назва аеропортових зборів, грошова сума.

Процеси: реалізувати інтерфейс користувача для додавання аеропортових зборів в підсистему.

- F1.5) Надає змогу користувачу підсистеми (економісту, адміністратору) вилучати аеропортові збори.

Процеси: реалізувати інтерфейс користувача для вилучення аеропортових зборів.

- F1.6) Надає змогу користувачу підсистеми (економісту, адміністратору) редагувати аеропортові збори.

Процеси: реалізувати інтерфейс користувача для редагування аеропортових зборів.

- F1.7) Надає змогу користувачу підсистеми (економісту, адміністратору) переглядати список тарифів.

Процеси: реалізувати інтерфейс користувача для перегляду тарифів у вигляді таблиці.

					ДР.122.4142.03.ПЗ.Р02	Аркуш
						46
Зм.	Аркуш	№ документа	Підпис	Дата		

- F1.8) Надає змогу користувачу підсистеми (економісту, адміністратору) переглядати список аеропортових зборів.

Процеси: реалізувати інтерфейс користувача для перегляду аеропортових зборів у вигляді таблиці.

- F2.1) Надає змогу користувачу (клієнту) придбати квиток на необхідний рейс та вибрану дату вильоту використовуючи сайт підсистеми.

Дані від клієнта: прізвище, ім'я, вік, номер телефону, електронна пошта, серія паспорта, номер паспорта, клас.

У разі адекватності всіх даних відбувається переадресація на сторінку оплати квитка(для оплати квитка використовується стороння банківська підсистема). Після придбання квиток надсилається на вказану електронну адресу.

Процеси: реалізувати інтерфейс користувача, який надає можливість придбання авіаквитків через сайт.

- F2.2) Надає змогу користувачу (клієнту) придбати квиток на необхідний рейс та дату вильоту використовуючи касу аеропорту. Клієнт надає необхідну інформацію касиру (дані від клієнта з пункту F2.1). У разі адекватності даних клієнт оплачує квиток. Після придбання квиток надсилається на вказану електронну адресу та надається клієнту в друкованому вигляді.

- F3.1) Надає змогу диспетчерській службі використовувати, оновлювати, сортувати (за різними атрибутами) розклад вильотів та прильотів. Диспетчерам надається повна інформація про польоти для повного контролю за польотами в аеропорту.

Процеси: реалізувати інтерфейс користувача, який надає можливість перегляду, сортування, оновлення розкладу вильотів та польотів у вигляді таблиці.



- F3.2) Надає змогу клієнтам (пасажирам), переглядати плановий розклад вильотів та прильотів літаків в аеропорту. Розклад для пасажирів формується диспетчерами кожні два тижні.

Процеси: реалізувати інтерфейс користувача, який надає можливість перегляду розкладу для потенційних пасажирів. Реалізувати можливість пошуку та сортування.

- F4.1.1) Надає змогу адміністратору додавати пілотів в підсистему (базу даних). Вхідні дані: номер посади, прізвище, ім'я, дата народження, адреса, телефон.

Процеси: реалізувати інтерфейс користувача для додавання нових пілотів.

- F4.1.2) Надає змогу адміністратору вилучати пілотів з підсистеми (базу даних).

Процеси: реалізувати інтерфейс користувача для вилучення пілотів.

- F4.1.3) Надає змогу адміністратори редагувати дані пілотів.

Процеси: реалізувати інтерфейс користувача для редагування даних пілотів.

- F4.1.4) Надає змогу адміністратору додавати посади пілотів в підсистему. Вхідні дані: назва посади, оклад, премія, стаж.

Процеси: реалізувати інтерфейс користувача для додавання посад.

- F4.1.5) Надає змогу адміністратору вилучати посади з підсистеми.

Процеси: реалізувати інтерфейс користувача для вилучення посад.

- F4.1.6) Надає змогу адміністратору редагувати дані посади.

Процеси: реалізувати інтерфейс користувача для редагування даних посад.

- F4.1.7) Надає змогу адміністратору переглядати всіх пілотів, які є у підсистемі.

Процеси: реалізувати інтерфейс користувача для перегляду пілотів у вигляді таблиці.

					ДР.122.4142.03.ПЗ.Р02	Аркуш
						48
Зм.	Аркуш	№ документа	Підпис	Дата		

- F4.1.8) Надає змогу адміністратору переглядати всі посади.  
Процеси: реалізувати інтерфейс користувача для перегляду посад у вигляді таблиці.
- F4.2.1) Надає змогу адміністратору додавати екіпажи в підсистему.  
Вхідні дані: назва екіпажу, кількість пілотів, стаж.  
Процеси: реалізувати інтерфейс користувача для додавання нових екіпажів в підсистему.
- F4.2.2) Надає змогу адміністратору вилучати екіпажи з підсистеми.  
Процеси: реалізувати інтерфейс користувача для вилучання екіпажів.
- F4.2.3) Надає змогу адміністратору редагувати дані екіпажів.  
Процеси: реалізувати інтерфейс користувача для редагування екіпажів.
- F4.2.4) Надає змогу адміністратору додавати пілотів в екіпаж.  
Процеси: реалізувати інтерфейс користувача для додавання пілотів в екіпаж.
- F4.2.5) Надає змогу адміністратору вилучати пілотів з екіпажу.  
Процеси: реалізувати інтерфейс користувача для вилучання пілотів з екіпажу.
- F4.2.6) Надає змогу адміністратору переглядати список всіх екіпажів  
Процеси: реалізувати інтерфейс користувача для перегляду списку екіпажів у вигляді таблиці.
- F4.3.1) Надає змогу адміністратору додавати нові літаки в підсистему.  
Вхідні дані: марка літака, авіакомпанія, максимальна швидкість, бортовий номер, кількість посадкових місць, паливо.  
Процеси: реалізувати інтерфейс користувача для додавання нових літаків.
- F4.3.2) Надає змогу адміністратору вилучати літаки з підсистеми.  
Процеси: реалізувати інтерфейс користувача для вилучання літаків.
- F4.3.3) Надає змогу адміністратору редагувати дані літаків.

Процеси: реалізувати інтерфейс користувача для редагування даних літаків.

- F4.3.4) Надає змогу адміністратору переглядати список всіх літаків.

Процеси: реалізувати інтерфейс користувача для перегляду списку літаків у вигляді таблиці.

- F4.4.1) Надає змогу адміністратору додавати нові рейси в підсистему.

Вхідні дані: назва рейсу, авіакомпанія, час вильоту, час прибуття, пункту відправлення, пункту призначення, тривалість рейсу.

Процеси: реалізувати інтерфейс користувача для додавання нових рейсів в підсистему.

- F4.4.2) Надає змогу адміністратору вилучати рейси з підсистеми.

Процеси: реалізувати інтерфейс користувача для вилучання рейсів.

- F4.4.3) Надає змогу адміністратору редагувати дані рейсів.

Процеси: реалізувати інтерфейс користувача для редагування даних рейсів.

- F4.4.4) Надає змогу адміністратору додавати нові пункти призначення та відправлення (аеропорти) в підсистему

Вхідні дані: Назва аеропорту, назва країни, назва міста.

Процеси: реалізувати інтерфейс користувача для додавання нових пунктів призначення та відправлення (аеропортів) в підсистему.

- F4.4.5) Надає змогу адміністратору вилучати пункти призначення та відправлення з підсистеми.

Процеси: реалізувати інтерфейс користувача для вилучання пунктів призначення та відправлення.

- F4.4.6) Надає змогу адміністратору редагувати дані пунктів призначення та відправлення.

Процеси: реалізувати інтерфейс користувача для редагування пунктів призначення та відправлення.

- F4.4.7) Надає змогу адміністратору переглядати всі рейси.

Процеси: реалізувати інтерфейс користувача для перегляду списку рейсів у вигляді таблиці.

- F4.4.8) Надає змогу адміністратору переглядати всі пункти призначення та відправлення.

Процеси: реалізувати інтерфейс користувача для перегляду списку пунктів призначення та відправлення у вигляді таблиці.

- F4.5.1) Надає змогу адміністратору додавати нові польоти (вильоти та прильоти) в підсистему.

Вхідні дані: рейс, екіпаж, літак, дата польоту, тариф, аеропортовий збір.

Процеси: реалізувати інтерфейс користувача для додавання польотів.

- F4.5.2) Надає змогу адміністратору вилучати польоти з підсистеми.

Процеси: реалізувати інтерфейс користувача для вилучення польотів.

- F4.5.3) Надає змогу адміністратору редагувати дані польотів,

Процеси: реалізувати інтерфейс користувача для редагування польотів.

- F4.5.4) Надає змогу адміністратору переглядати всі польоти (вильоти та прильоти).

- Реалізувати інтерфейс користувача для перегляду списку польотів у вигляді таблиці. Реалізувати можливість перегляду списку вильотів окремо, польотів окремо та можливість пошуку, сортування.

- F5.1) Формує список зареєстрованих багажів на певний виліт. Надає можливість перегляду списку співробітникам відділу, які займаються завантажуванням і розвантажуванням багажів в вантажні відділи літаків.

Процеси: реалізувати інтерфейс користувача, який надає можливість перегляду списку зареєстрованих багажів на певний виліт. Реалізувати можливість відправлення списку електронною поштою.

- F5.2) Формує список пасажирів на певний виліт. Надає можливість перегляду списку співробітникам відділу, які займаються процесом посадки і прийому пасажирів.

Процеси: реалізувати інтерфейс користувача, який надає можливість перегляду списку пасажирів на певний виліт. Реалізувати можливість відправлення списку електронною поштою.

- F6.1) Надає змогу клієнту (пасажиру) зареєструвати та здати свій багаж в аеропорту. Багаж надається співробітникам аеропорту, де його зважать, та занесуть інформацію в базу даних.

Процеси: реалізувати інтерфейс користувача для додавання багажів.

- F6.2) Надає змогу співробітникам аеропорту вилучати багаж з підсистеми.

Процеси: реалізувати інтерфейс користувача для вилучення багажу.

- F6.3) Надає змогу співробітникам аеропорту редагувати дані багажу.

Процеси: реалізувати інтерфейс користувача для редагування даних багажу.

Процеси: реалізувати інтерфейс користувача, який надає можливість додавання, вилучення багажу з бази даних підсистеми.

- F7) Надає можливість зареєструвати пасажирів на виліт. Реєстрація проходить в аеропорті за починається за три години до вильоту Для реєстрації пасажир надає номер квитка та паспортні дані. В результаті успішної реєстрації пасажир отримує посадковий талон із зазначенням місця.

- F8) Надає змогу адміністраторам підсистеми розподіляти права доступу між користувачами підсистеми.

Процеси: реалізувати інтерфейс користувача, який надає можливість виконувати розподілення ролей між користувачами.

- F9) Надає змогу користувачу зареєструватися в підсистемі. Вхідні дані: логін, пароль, електронна пошта.

Процеси: реалізувати інтерфейс користувача, який надає можливість зареєструватися в підсистемі.

- F10) Надає змогу користувачу авторизуватися в підсистемі. Вхідні дані: логін або електронна пошта , пароль.

Процеси: реалізувати інтерфейс користувача, який надає можливість авторизуватися в підсистемі.

### 2.1.3. Загальний опис підсистеми

Підсистема “Обслуговування пасажирів” ІС аеропорту призначена для відділу аеропорту з обслуговування пасажирів, користувачами даної підсистеми є клієнти (пасажирів), касири, адміністратори, диспетчери, та інші співробітники аеропорту.

Підсистема автоматизує процеси обробки інформації, забезпечує оперативну передачу та отримання даних, безпечний доступ до даних та зберігання даних, розмежування прав доступу до підсистеми, підвищує ефективність роботи персоналу аеропорту реалізує служби: з продажу квитків, реєстрації багажу, формування розкладів прийомів та відправлень літаків міжнародних та внутрішніх рейсів.

Основними функціональними характеристиками підсистеми є:

- реєстрація користувача;
- авторизація користувача;
- управління правами доступу;
- продаж квитків на рейс та необхідну пасажиру дату вильоту за допомогою веб-додатку ;
- продаж квитків, використовуючи касу аеропорту;
- автоматизація процесу формування ціни на авіаквитки;
- формування розкладу для пасажирів - надає клієнтам (пасажирам) можливість перегляду розкладу вильотів та прильотів літаків за допомогою веб-додатку підсистеми;

- формування розкладу для диспетчерської служби – надає можливість диспетчерам формувати, переглядати, редагувати розклад як вони вважають за потрібне для себе
- реєстрація багажу пасажирів
- формування, надання та перегляд списків пасажирів на вильоти;
- формування, надання та перегляд списків багажів на вильоти;
- адміністрування бази даних підсистеми, реалізує можливість перегляду, додавання, видалення, редагування інформації;

Підсистема має зручний, динамічний та інтуїтивно зрозумілий інтерфейс користувача, що полегшує та пришвидшує її використання.

#### **2.1.4. Діаграма використання**

Діаграма прецедентів — UML діаграма, на якій зображено відношення між акторами та прецедентами в системі. Також, перекладається як діаграма варіантів використання. Актори на діаграмі варіантів використання позначаються символом людини, а варіанти використання – еліпсом.

Суть даної діаграми полягає в наступному: проєктована система представляється у вигляді безлічі сутностей чи акторів, що взаємодіють із системою за допомогою, так званих, варіантів використання. Варіант використання використовують для описання послуг, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система при діалозі з актором. При цьому нічого не говориться про те, яким чином буде реалізована взаємодія акторів із системою [5].

У мові UML є кілька стандартних видів відношень між акторами і варіантами використання:

- асоціації;
- включення;
- розширення;
- узагальнення.

Діаграма варіантів використання не авторизованого користувача зображена на рисунку 2.3.

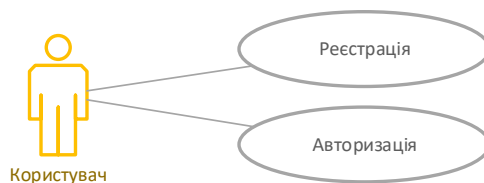


Рисунок 2.3 – Діаграма використання не авторизованого користувача

Діаграма варіантів використання підсистеми після авторизації зображена на рисунку 2.4.

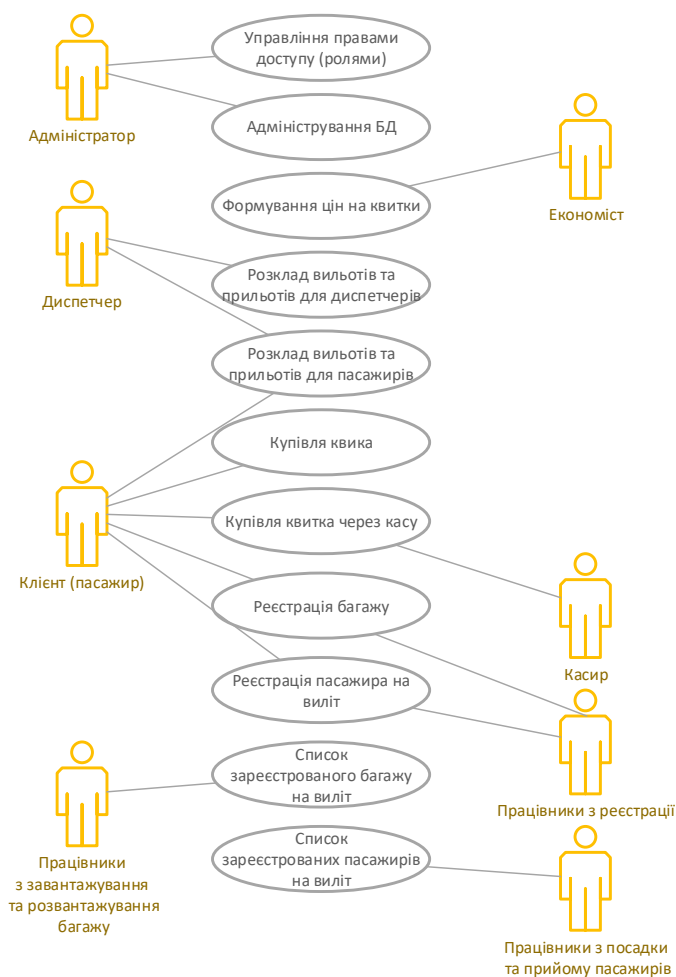


Рисунок 2.4 – Діаграма варіантів використання підсистеми після авторизації користувача



Декомпозиція варіанту використання «Формування ціни на квитки» представлена на рисунку 2.5.



Рисунок 2.5 - Діаграма декомпозиції варіанту використання «Формування ціни на квитки»

Декомпозиція варіанту використання «Реєстрація багажу» представлена на рисунку 2.6



Рисунок 2.6 - Діаграма декомпозиції варіанту використання «Реєстрація багажу»

Декомпозиція варіанту використання «Адміністрування БД» представлена на рисунку 2.7.



Рисунок 2.7 - Діаграма декомпозиції варіанту використання «Адміністрування БД»

Декомпозиція варіанту використання «Адміністрування польотів» представлена на рисунку 2.8.

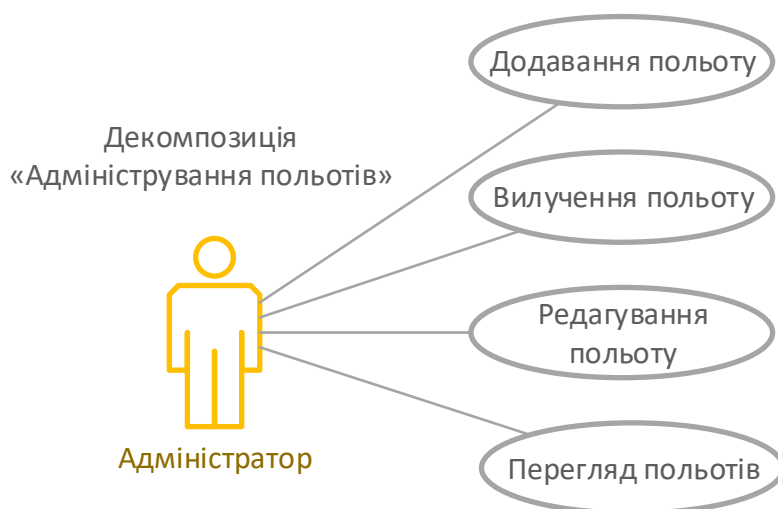


Рисунок 2.8 - Діаграма декомпозиції варіанту використання «Адміністрування польотів»

Декомпозиція варіанту використання «Адміністрування рейсів» представлена на рисунку 2.9.

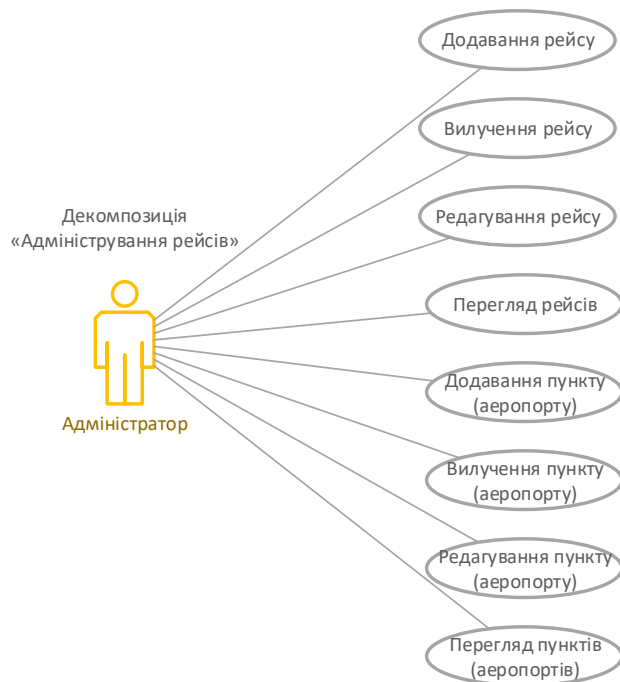


Рисунок 2.9 - Діаграма декомпозиції варіанту використання «Адміністрування рейсів»

Декомпозиція варіанту використання «Адміністрування екіпажів» представлена на рисунку 2.10.

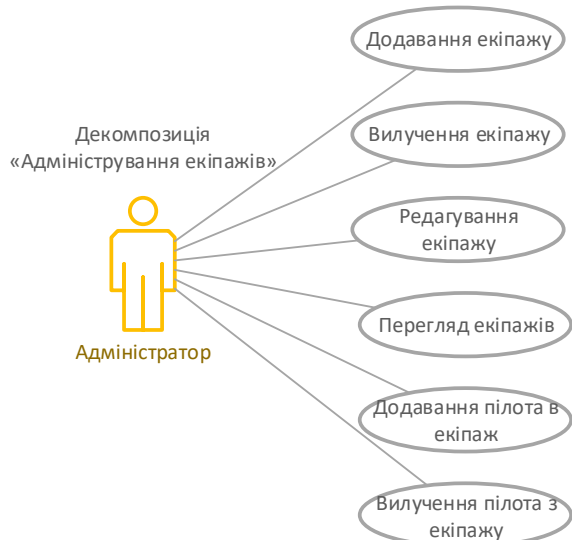


Рисунок 2.10 - Діаграма декомпозиції варіанту використання «Адміністрування екіпажів»

Декомпозиція варіанту використання «Адміністрування пілотів» представлена на рисунку 2.11.



Рисунок 2.11 - Діаграма декомпозиції варіанту використання «Адміністрування пілотів»

Декомпозиція варіанту використання «Адміністрування літаків» представлена на рисунку 2.12.

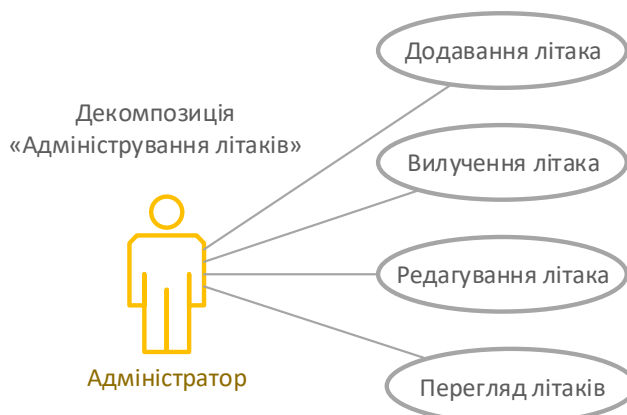


Рисунок 2.12 - Діаграма декомпозиції варіанту використання «Адміністрування літаків»

### 2.1.5. Діаграма діяльності

Діаграма діяльності – в UML, візуальне представлення графу діяльностей. Граф діяльностей є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій.

Дія є фундаментальною одиницею визначення поведінки в специфікації. Дія отримує множину вхідних сигналів, та перетворює їх на множину вихідних сигналів. Одна із цих множин, або обидві водночас, можуть бути порожніми. Виконання дії відповідає виконанню окремої дії. Подібно до цього, виконання діяльності є виконанням окремої діяльності, буквально, включно із виконанням тих дій, що містяться в діяльності. Кожна дія в діяльності може виконуватись один, два, або більше разів під час одного виконання діяльності. Щонайменше, дії мають отримувати дані, перетворювати їх та тестувати, деякі дії можуть вимагати певної послідовності. Специфікація діяльності може дозволяти виконання декількох (логічних) потоків, та існування механізмів синхронізації для гарантування виконання дій у правильному порядку.

Діаграми активностей будуються з обмеженої кількості фігур, з'єднаних стрілочками. Найважливіші типи фігур:

- скруглені прямокутники позначають дії;
- ромби позначають рішення;
- синій кружок позначає старт (початковий стан) процесу;
- зелений кружок в колі позначає кінець (кінцевий стан).

Стрілки ведуть від старту до кінця і позначають порядок в якому відбуваються активності.

Діаграма діяльності користувача з роллю клієнта (пасажира) при роботі з підсистемою представлена на рисунку 2.13.

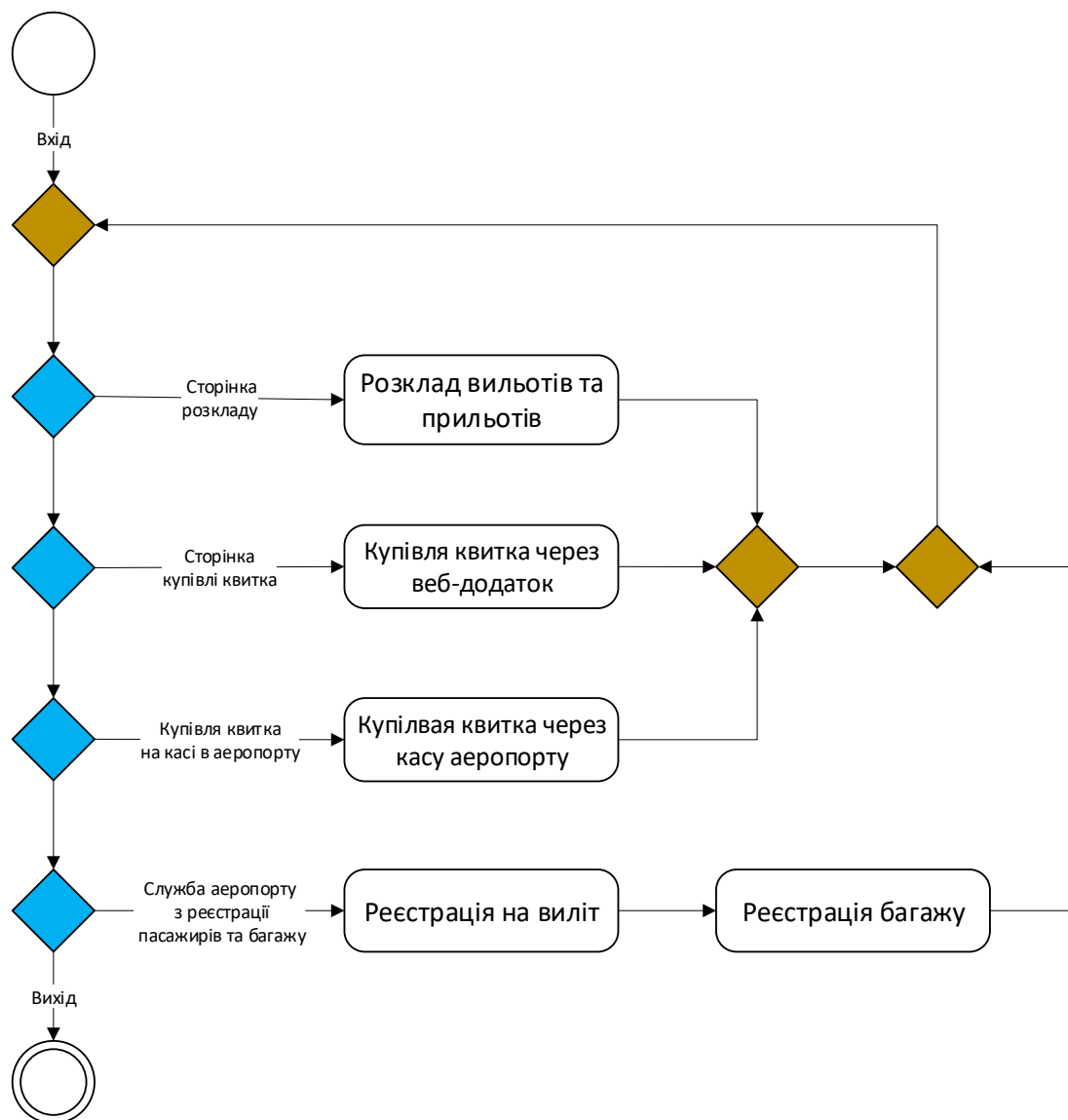


Рисунок 2.13 - Діаграма діяльності клієнта (пасажира) аеропорту.

Діаграма діяльності адміністратора при роботі з підсистемою представлена на рисунку 2.14.

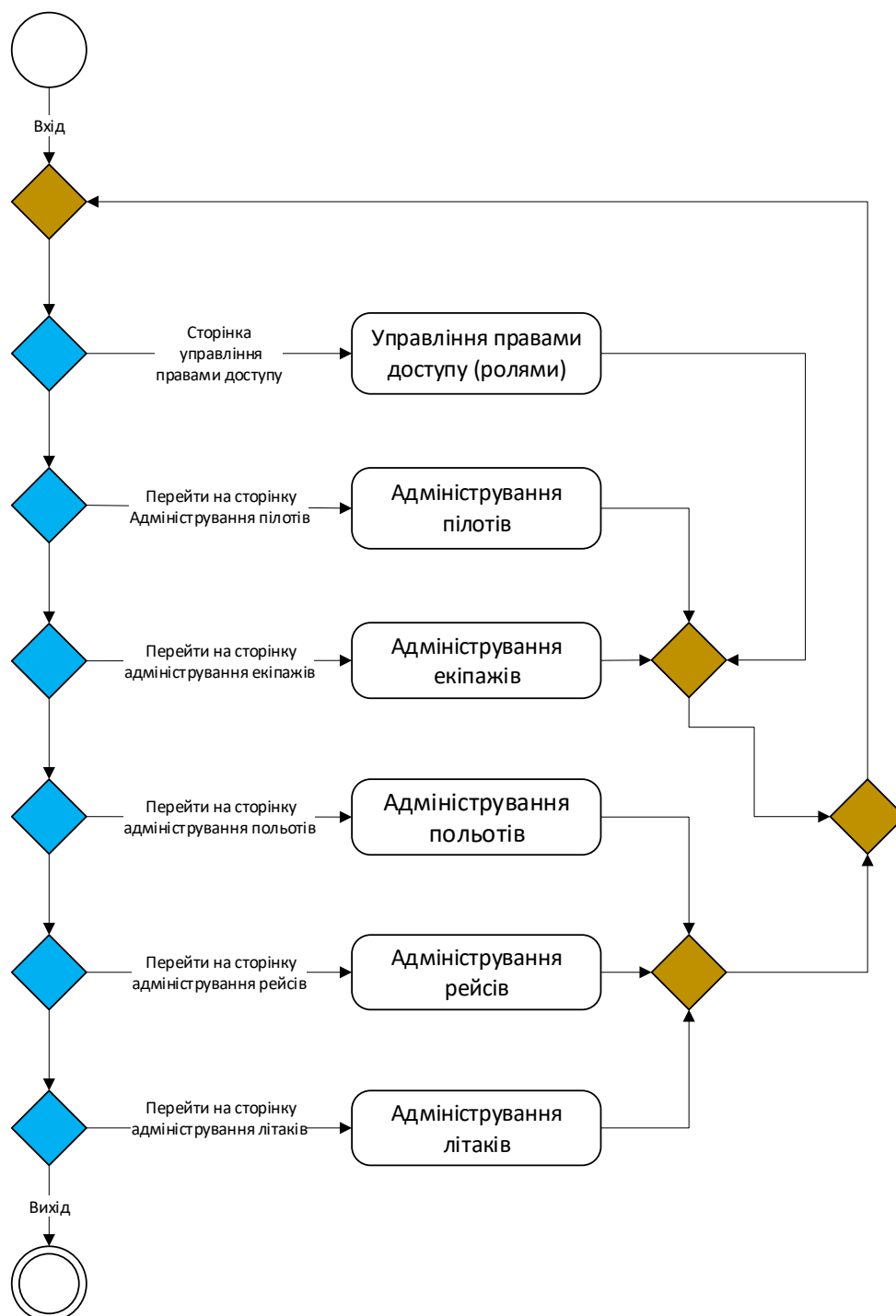


Рисунок 2.14 - Діаграма діяльності адміністратора аеропорту.

## **2.2. Рішення з інформаційного забезпечення підсистеми**

### **2.2.1. Концептуальна та логічна моделі бази даних**

Після визначення функціональних вимог до підсистеми та її меж, слід проаналізувати предметну область з метою побудови діаграми класів, яка складає визначення внутрішнього стану підсистеми. На даному буде визначено постійні сутності бізнесів-процесів. Бізнес-процес - стійка, цілеспрямована сукупність взаємопов'язаних видів діяльності (послідовність робіт), яка за певною технологією перетворює вхідний продукт на вихідний, що представляють цінність для внутрішнього або зовнішнього споживача. Вид класів, що забезпечують опис бізнес процесів назовемо класами-сутностями або класами предметної області.

Одним із способів змалювання внутрішнього стану системи є створення концептуальної та логічної моделі бази даних.

Концептуальна модель визначає вимоги користувачів бази даних, які оформлені у вигляді таблиць, схем інформаційних потоків, схем взаємозв'язків між показниками техніко-економічної інформації, порядку їх розрахунку та коментарів до них. Таким чином, проектування концептуальної моделі засновано на аналізі задач обробки даних, що мають розв'язуватися в інформаційній системі [7].

					ДР.122.4142.03.ПЗ.Р02	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		<b>63</b>



Концептуальна модель даних підсистеми “Обслуговування пасажирів” ІС аеропорту представлена на рисунку 2.15.

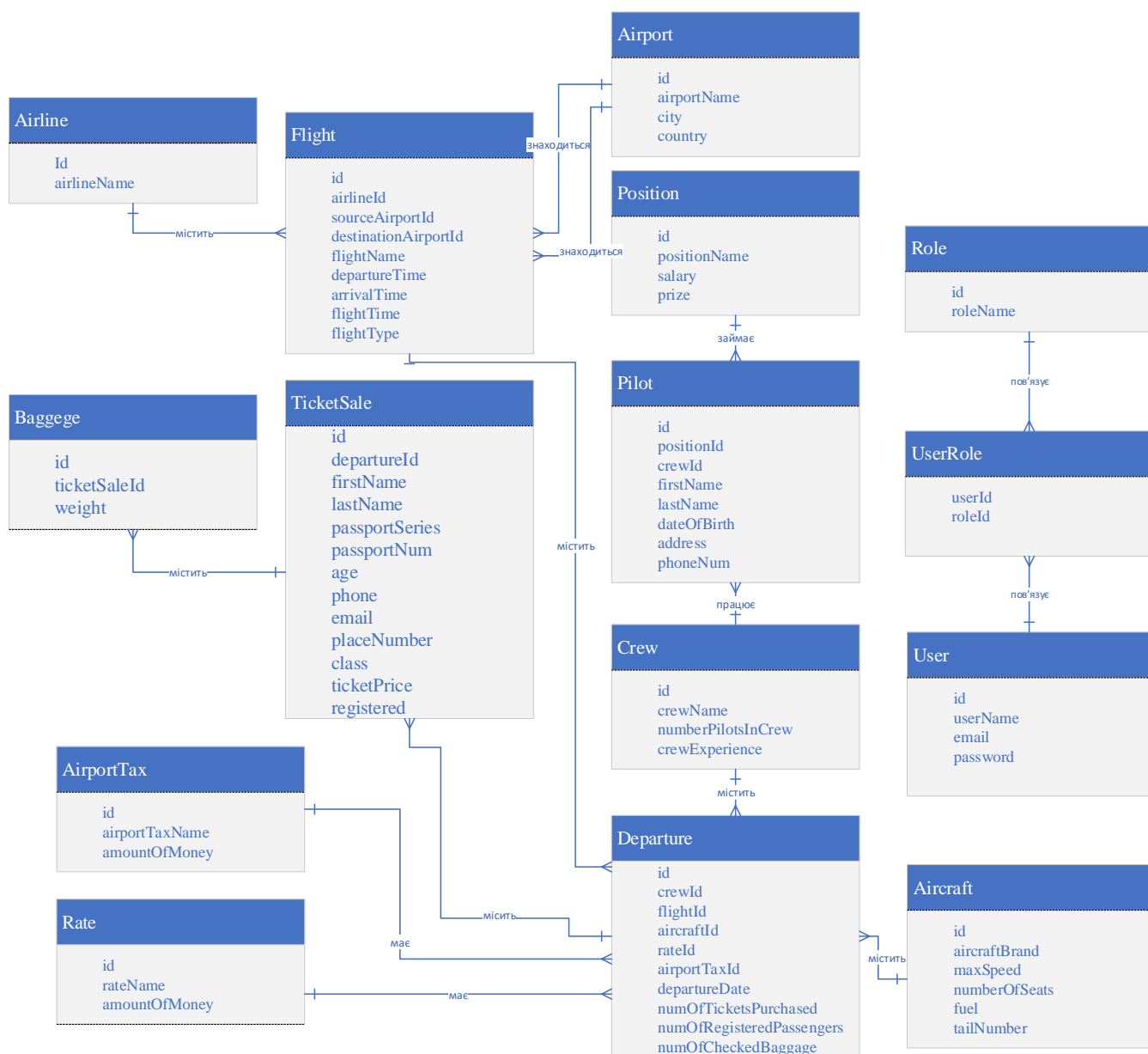


Рисунок 2.15 – Концептуальна модель даних

Логічна модель даних — модель даних конкретної предметної області, виражена незалежно від конкретного продукту керування базами даних або технології зберігання (фізична модель даних), але в термінах структур даних, таких як реляційні таблиці та колонки, об'єктно-орієнтовані класи чи теги XML. Вона є протилежністю концептуальній моделі даних, яка описує семантику організації без посилання на технологію.

На основі представленої концептуальної моделі спроектовано логічну модель даних предметної області.

Логічна модель даних підсистеми “Обслуговування пасажирів” ІС аеропорту представлена на рисунку 2.16.

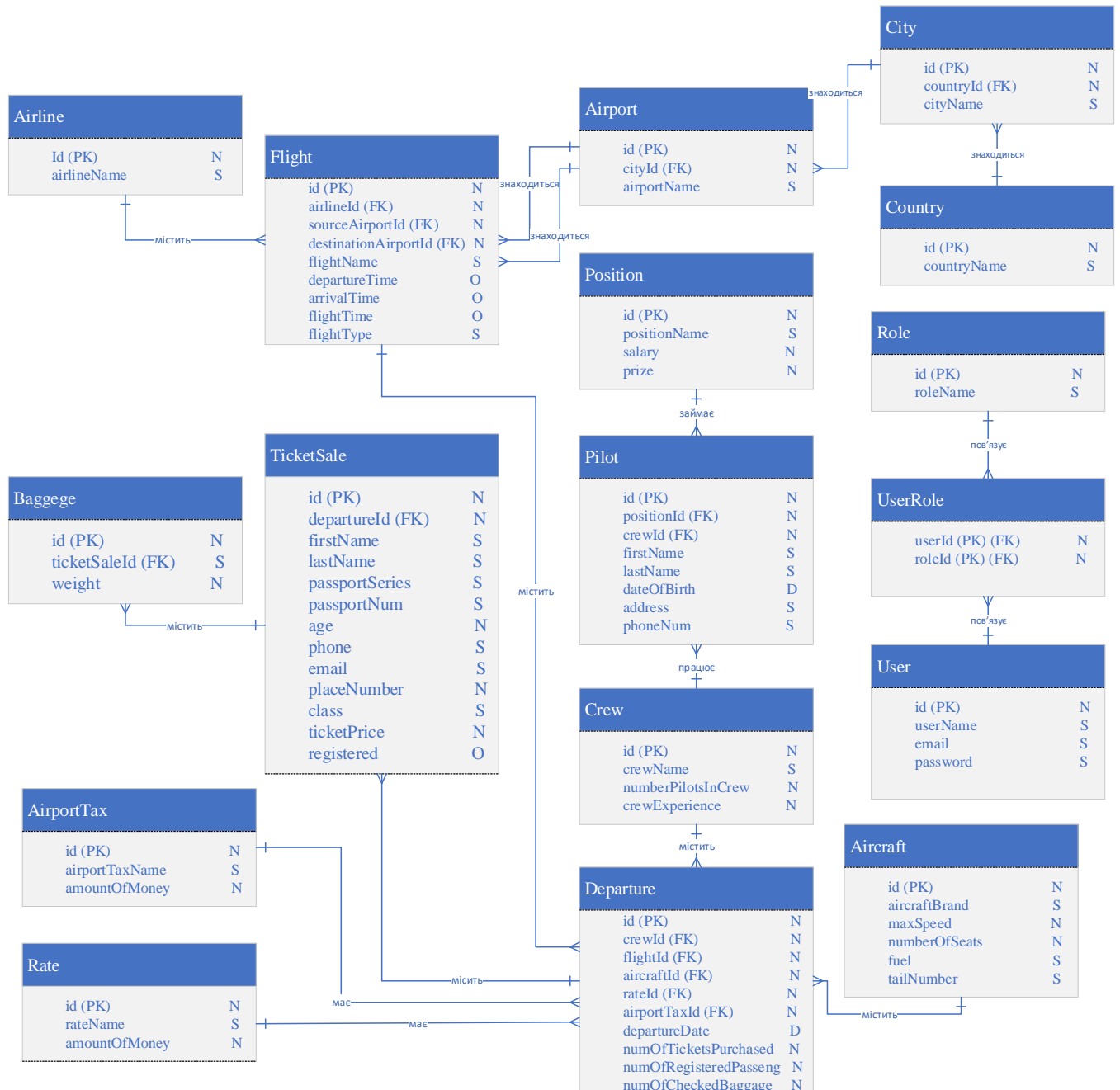


Рисунок 2.16 – Логічна модель даних

На даному етапі сутності моделі даних стають кандидатами для класів проєктованої підсистеми.

В діаграмі логічної моделі даних зображено наступні сутності:

- Airline – інформація про авіакомпанію
- Airport – інформація про пункт (аеропорт) призначення або відправлення.
- City – інформація про місто знаходження аеропорту;
- Country – інформація про країну знаходження аеропорту;
- Position – інформація про посаду пілота;
- Crew – інформація про екіпаж;
- AirportTax – інформація про аеропортовий;
- Rate – інформація про тариф;
- Aircraft – інформація про літак;
- TicketSale – інформація про продаж квитка;
- Baggage – інформація про багаж;
- Flight – інформація про рейс;
- Pilot – інформація про пілота;
- Departure – інформація про політ;
- User – інформація про користувача;
- Role – інформація про роль;
- UserRole – зв'язна таблиця між User і Role

### 2.2.2 Аналіз нормалізації сутностей бази даних:

Аналіз нормалізації сутностей бази даних:

- Жоден атрибут не містить багатослівних частин інформації (крім атрибуту адреса (address) в таблиці пілот (pilot). Цей атрибут буде використовуватися тільки для зберігання адресу пілота та не може бути розділений на декілька частин, а отже є атомарним. Крім того в таблицях не зустрічаються повторення даних з чого зроблено висновок, що всі таблиці БД знаходяться у першій нормальній формі.
- Так як всі відношення у БД знаходяться у першій нормальній формі та кожен не ключовий атрибут повністю залежить від первинного ключа,

					ДР.122.4142.03.ПЗ.Р02	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		66

котрий для кожної таблиці є простим (складається з одного атрибуту), то усі відношення автоматично знаходяться у другій нормальній формі.

- Розглядаючи не ключові атрибути у таблицях БД зроблено висновок, що вони залежать винятково від первинного ключа, а оскільки всі відношення знаходяться у другій нормальній формі то всі вони знаходяться в третій нормальній формі.

Всі таблиці представлені раніше знаходяться у третій нормальній формі.

## **2.3. Рішення з програмного забезпечення підсистеми**

### **2.3.1. Структура програмного забезпечення**

Діаграма класів призначена для надання статичної структури моделі системи в термінології класів об'єктно-орієнтованого програмування. Діаграма класів відображує різні взаємозв'язки між окремими сутностями предметної області, такими як об'єкти й підсистеми, а також описує їхню внутрішню структуру й типи відносин. На даній діаграмі не вказується інформація про часові аспекти функціонування системи.

Діаграма класів (class diagram) - діаграма на якій представлена сукупність декларативних або статичних елементів моделі, таких як класи з атрибутами й операціями, а також відношення, що їх з'єднують.

Клас (class) - абстрактний опис множини однорідних об'єктів, що мають однакові атрибути, операції й відносини з об'єктами інших класів. Графічно клас зображується у вигляді прямокутника, що додатково може бути розділений горизонтальними лініями на розділи або секції. У цих секціях можуть вказуватися ім'я класу, атрибути й операції класу [17].

Діаграма основних класів підсистеми представлена на рисунку 2.17

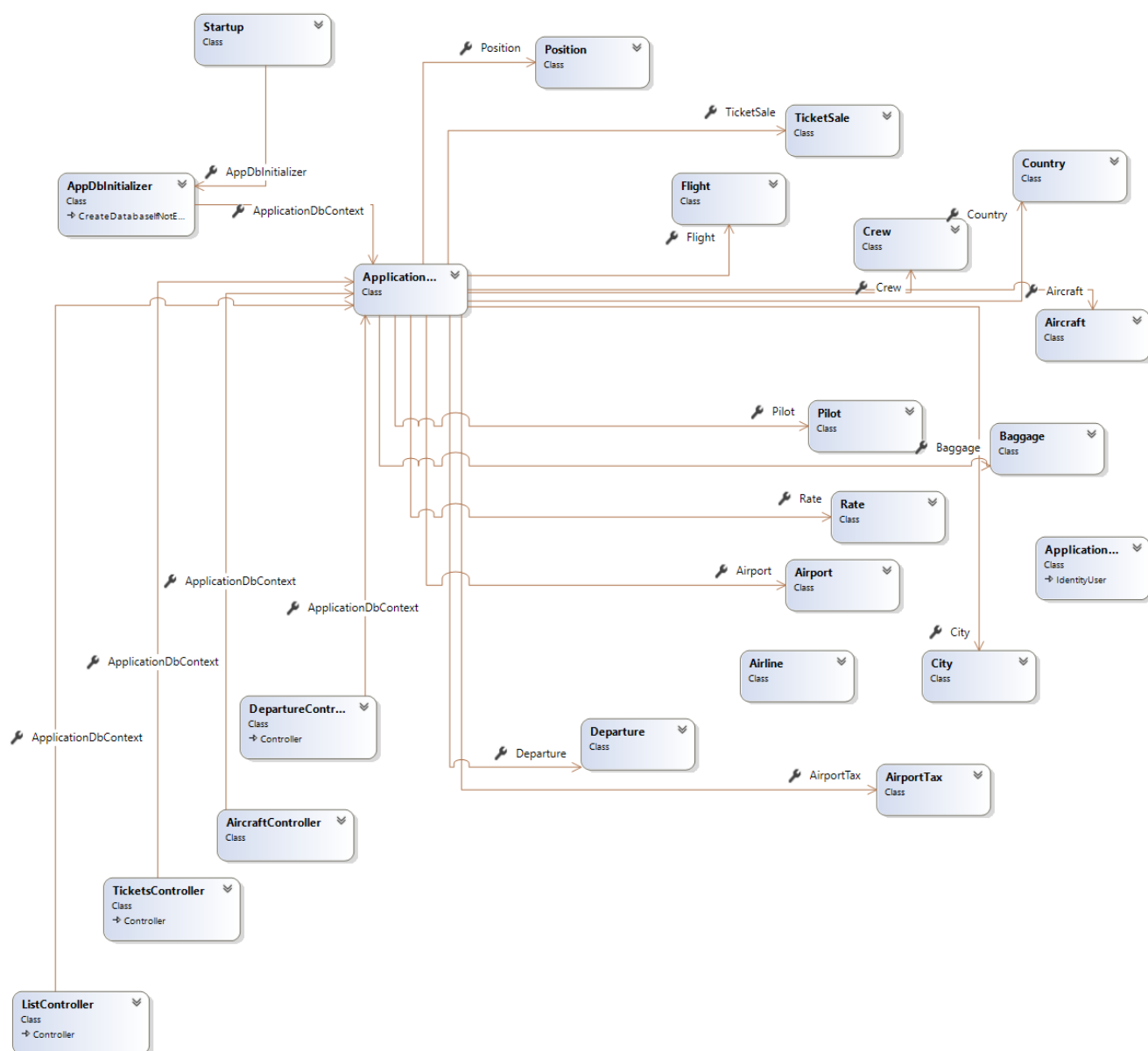


Рисунок 2.17 – Діаграма основних класів

### 2.3.2. Фізична модель бази даних

Для забезпечення можливості зберігання даних в реляційній базі даних, з моделі класів була отримана фізична модель даних.

Фізична модель даних підсистеми “Обслуговування пасажирів” ІС аеропорту представлена на рисунку 2.18.

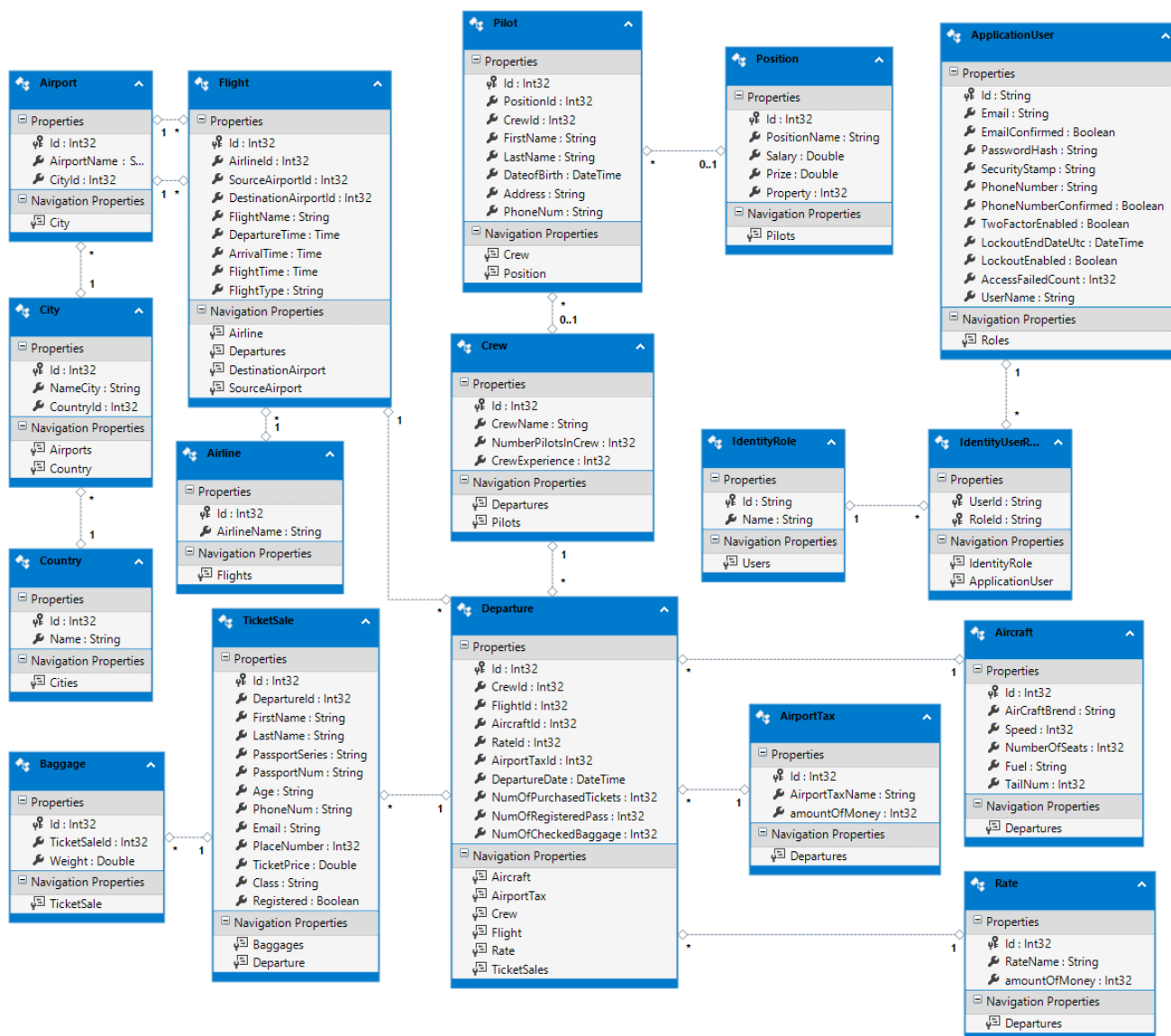


Рисунок 2.18 – Фізична модель даних

Фізична модель даних — подання дизайну даних, як реалізованого чи призначеного для реалізації у системі керування базами даних. Завершена фізична модель даних включає всі артефакти бази даних, необхідні для створення відношень між таблицями чи для досягнення мети продуктивності, як-от індексів, визначень обмежень і зв'язаних таблиць або кластерів. Для реалізації моделі бази даних була обрана СКБД MSSQL [7].

Робота с базою даних відбувається за допомогою Entity Framework. Цей клас надає методи для зручної роботи з об'єктами, який значно підвищить продуктивність.

### 2.3.3. Методи і засоби розробки програмного забезпечення

При постановці вимог, яким повинна відповідати підсистема та проектуванні було застосовано об'єктно-орієнтований підхід, тому реалізація підсистеми повинна дотримуватися саме цього підходу. Такий підхід має низку переваг і головні можна виділити наступні:

Використання ООП у розробці дозволяє:

- Класи дозволяють проводити конструювання з корисних компонент, що володіють простими інструментами, що дає можливість абстрагуватися від деталей реалізації.
- Дані та операції разом утворюють певну сутність і вони не «розмазуються» по всій програмі, як це нерідко буває в разі процедурного програмування.
- Локалізація коду і даних покращує наочність і зручність супроводу програмного забезпечення.
- Інкапсуляція інформації захищає найбільш критичні дані від несанкціонованого доступу.
- Уникнення дублювання коду. Це дозволяє уникати помилок в коді, інтерфейс стає більш уніфікованим.

Засобом розробки програмного забезпечення клієнтського рівня підсистеми було обрано такі технології HTML, CSS, Bootstrap 4, JavaScript. Ці технології є найбільш поширеними при розробці верстки та дизайну інтернет сторінок, а також були розглянуті в рамках навчального курсу.

Засобом розробки програмного забезпечення серверного рівня підсистеми є мова програмування високого рівня C#, та такі технології як: ASP.NET MVC

5, Entity Framework, LINQ, ASP.NET Identity. У якості середовища розробки використано продукт Microsoft Visual Studio 2019.

На рівні серверу баз даних використовується вільна система керування реляційними базами даних MSSQL Server, це сучасна серверна реляційна СКБД, що має високу продуктивність, здатна працювати з великими обсягами інформації.

Переваги через які було обрано всі перелічені засоби описані в першому розділі роботи.

## **2.4. Діаграма розгортання**

Варіантами архітектури підсистеми, що задовольняють поставленим вимогам, є клієнт-серверна та сервіс-орієнтована архітектури.

За трирівневою клієнт-серверною архітектурою передбачається наявність наступних компонентів програми: клієнтський застосунок, підключений до сервера застосунків, який в свою чергу підключений до серверу бази даних. Така архітектура передбачає ізолюваність рівнів один від одного що дозволяє швидко і простими засобами переконфігурувати систему при виникненні збоїв або при плановому обслуговуванні на одному з рівнів. Крім того є масштабованою, надійною та має високий рівень безпеки.

Сервіс-орієнтована архітектура є модульним підходом до розробки систем, заснованим на використанні сервісів (служб) зі стандартизованими інтерфейсами. В її основі лежать принципи багатократного використання функціональних елементів ІТ, ліквідації дублювання функціональності в ПЗ, уніфікації типових операційних процесів тощо. Компоненти такої системи можуть бути розподілені по різних вузлах мережі, і пропонуються як незалежні, слабо пов'язані, замінні сервіси-додатки. Інтерфейс компонентів такої системи надає інкапсуляцію деталей реалізації конкретного компонента (ОС, платформи, мови програмування, розробника тощо).



Клієнт-серверна архітектура є більш зручною з боку використання, адже для доступу до системи (сервер) не потрібно специфічного програмного забезпечення, достатньо використання вбудованих можливостей операційних систем – браузерів (клієнт). Тому обираємо цей підхід як підхід до вирішення завдання розробки системи прогнозування обсягу програмного забезпечення.

Згідно з трирівневою клієнт-серверною архітектурою, підсистема повинна бути поділена на три рівні. Для розгортання кожного з рівнів будуть використовуватися наступні компоненти:

Клієнтський рівень – браузер кінцевого користувача.

Серверний рівень – IIS (Internet Information Services) – набір сервісів від компанії Microsoft для роботи веб-сервера та інших інтернет служб. Основним компонентом IIS є веб-сервер, який надає клієнтам доступ до сайтів за протоколами HTTP і HTTPS.

SQL Server – як сервер баз даних.

Діаграма розгортання — діаграма, на якій відображаються обчислювальні вузли під час роботи програми, компоненти, та об'єкти, що виконуються на цих вузлах. Компоненти відповідають представленню робочих екземплярів одиниць коду. Діаграма розгортання підсистеми зображена на рисунку 2.19.

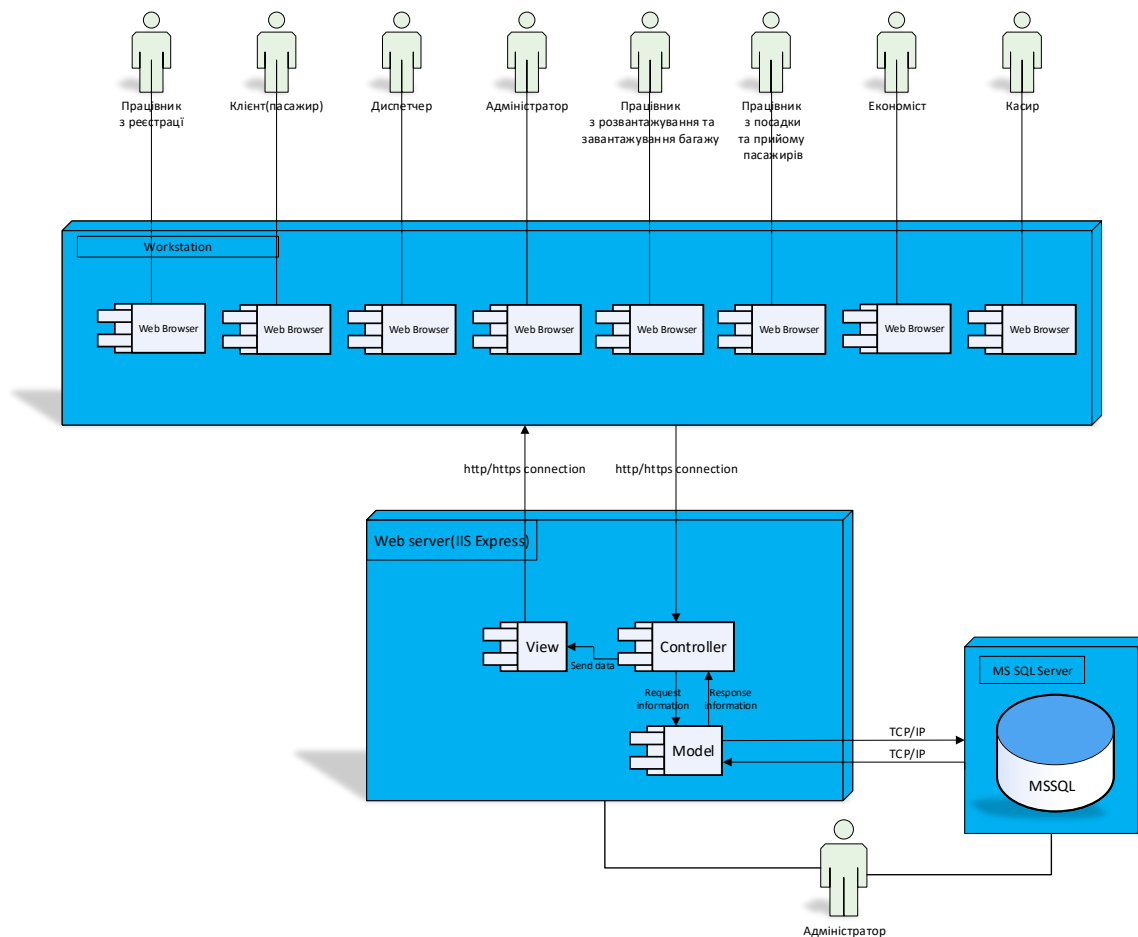


Рисунок 2.19 – Діаграма розгортання підсистеми

## 2.5. Висновки

В результаті розробки технічного проекту розроблено діаграми використання та діяльності, які наглядно відображають, які функції необхідно реалізувати в підсистемі та хто буде взаємодіяти з ними. Розроблено концептуальну модель бази даних, яка необхідна для опису зв'язків між об'єктами даних та їх характеристиками та логічну модель бази даних, яка стала основою для моделювання фізичної моделі бази даних, яка є остаточним варіантом представлення структури бази даних.

В цьому розділі було описано загальносистемні рішення, рішення з інформаційного забезпечення та рішення з програмного забезпечення. Ці рішення є інструкцією з реалізації підсистеми.

					ДР.122.4142.03.ПЗ.Р03			
Зм.	Аркаш	№ документа	Підпис	Дата				
					Реалізація підсистеми “Обслуговування пасажирів” ІС аеропорту	Літ.	Аркуш	Аркушів
Студент	Борис А. В.						74	14
Керівник	Маршак О. І.					НУК		
Зав. каф.	Михелєв І. Л.							

### 3. РЕАЛІЗАЦІЯ ПІДСИСТЕМИ “ОБСЛУГОВУВАННЯ ПАСАЖИРІВ” ІС АЕРОПОРТУ

#### 3.1. Технічне завдання

Технічне завдання для реалізованого проекту розроблене і представлене в додатку А.

#### 3.2. Опис розробки підсистеми

Програмне забезпечення створено за допомогою мови С# з використанням можливостей фреймворків ASP.NET MVC 5, Entity Framework, ASP.NET Identity.

С# є одним з лідерів серед мов програмування, об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET. Розроблена Андерсом Гейлсбергом, Скотом Вілтамутом та Пітером Гольде під егідою Microsoft Research.

Програмний продукт реалізовано за допомогою патерну MVC (Model View Controller).

Основна мета застосування цього патерну полягає в відділенні бізнес-логіки (моделі) від її візуалізації (уявлення, виду). За рахунок такого поділу підвищується можливість повторного використання коду. Найбільш корисне застосування даної концепції в тих випадках, коли користувач повинен бачити ті ж самі дані одночасно в різних контекстах. Зокрема, виконуються наступні завдання:

- до однієї моделі можна приєднати кілька видів, при цьому не зачіпаючи реалізацію моделі.
- не торкаючись реалізації видів, можна змінити реакції на дії користувача (натискання мишею на кнопки, введення даних) - для цього досить використовувати інший контролер;
- ряд розробників спеціалізується тільки в одній з областей: або розробляють графічний інтерфейс, або розробляють бізнес-логіку.

Концепція MVC надає можливість програмістам, які займаються

					ДР.122.4142.03.ПЗ.Р03	Аркуш
						75
Зм.	Аркуш	№ документа	Підпис	Дата		

розробкою бізнес-логіки (моделі), взагалі не знати, яке уявлення буде використовуватися.

Для розробки інтерфейсу користувача було використано HTTP (HyperText Transfer Protocol) протокол, що зумовлено його універсальністю та підтримкою більшої платформи. Для використання протоколу HTTP не потрібно стороннього програмного забезпечення крім браузера.

Для побудови веб-інтерфейсу використано мову розмітки HTML разом з каскадними таблицями стилів CSS та Bootstrap.

В якості бази даних використано СКБД MSSQL.

Для побудови таблиць бази даних було використано підхід Code First. За цим підходом спочатку визначаються моделі в коді, а потім на основі моделей, за допомогою можливостей фреймворку Entity Framework генерується скрипт створення бази даних, який представлено в додатку Б.

Текст програмних модулів підсистеми розроблено і представлено в додатку В.

### 3.3. Інструкція з використання підсистеми

Домашня сторінка підсистеми представлена на рисунку 3.1.

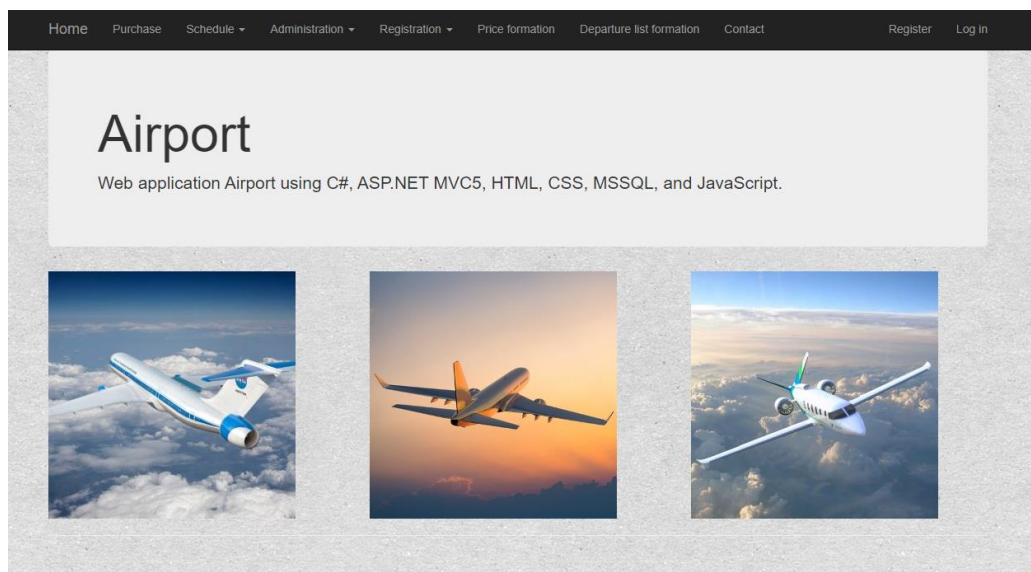


Рисунок 3.1 – домашня сторінка

Навігаційна панель веб-додатку включає такі пересилання (функції підсистеми):

- Авторизація (Log in);
- Реєстрація (Register);
- Купівля квитка (Purchase);
- Розклад польотів для пасажирів (Departure schedule);
- Розклад польотів для диспетчерів (Departure schedule for dispatchers);
- Адміністрування пілотів (Pilot administration);
- Адміністрування екіпажів (Crew administration);
- Адміністрування рейсів (Flight administration);
- Адміністрування вильотів (Departure administration);
- Адміністрування літаків (Aircraft administration);
- Формування цін (Price formation);
- Формування списку багажу та пасажирів на виліт (Departure list formation);
- Контактна інформація (Contact);
- Сторінка управління ролями (Role administration)

Сторінку реєстрації надає можливість зареєструватися в підсистемі, але це не обов'язкова процедура для клієнта (пасажир аеропорту). Після реєстрації користувачу надається роль клієнта (пасажир).

Сторінка реєстрації представлена на рисунку 3.2.

					ДР.122.4142.03.ПЗ.Р03	Аркуш
						77
Зм.	Аркуш	№ документа	Підпис	Дата		

Рисунок 3.2 – сторінка реєстрації

Для входження в підсистему треба перейти на сторінку вхід в підсистему.

Після входження в підсистему користувачу с роллю клієнт доступні наступні сторінки: купівлі квитка, розкладу вильотів для пасажирів, контактної інформації. Також ці сторінки доступні незареєстрованому користувачу.

Сторінка входу в підсистему (авторизації) представлена на рисунку 3.3.

Рисунок 3.3 – сторінка авторизації

Сторінка купівлі квитка представлена на рисунку 3.4.

Рисунок 3.4 – сторінка купівлі квитка

Для купівлі квитка потенційному пасажирові треба вибрати рейс, дату вильоту, ввести персональну інформацію і натиснути на кнопку Buy now. В разі вірності всіх введених даних буде пере направлено на сторінку інформація про купівлю, також ця інформація буде відправлена на вказану електронну пошту.

Сторінка успішної купівлі представлена на рисунку 3.5.

Рисунок 3.5 – сторінка успішної купівлі квитка



Розклад польотів для пасажирів представлено на рисунку 3.6.

Home	Purchase	Schedule ▾	Administration ▾	Registration ▾	Price formation	Departure list formation	Contact	Register	Log in
------	----------	------------	------------------	----------------	-----------------	--------------------------	---------	----------	--------

Schedule								
Flight	Please select ▾	Date	02/07/2020 ▾	Type	Please select ▾	Schedule		
Flight	Aircraft	Airline	Crew	Departure date	Departure time	Arrival time	Flight time	Flight type
Kiev - Saint petersburg	Beechcraft	Atlantic Airlines	Ninth crew	2020-07-02	11:00:00	15:00:00	04:00:00	Direct flight
Kiev - Charkow	Boeing	Air Europa	First crew	2020-07-02	07:00:00	09:00:00	02:00:00	Direct flight
Kiev - Athens	Cessna Aircraft	Air Europa	Seventeenth crew	2020-07-02	12:20:00	15:00:00	02:40:00	Direct flight
Kiev - Cologne	Boeing	Air Berlin	Second crew	2020-07-02	14:00:00	17:00:00	03:00:00	Direct flight
Kiev - Marseilles	Bombardier Aerospace	Air France	Fourteenth crew	2020-07-02	16:30:00	21:00:00	04:30:00	Direct flight
Kiev - Madrid	Cessna Aircraft	Air Europa	Eighteenth crew	2020-07-02	08:20:00	13:00:00	04:40:00	Direct flight
Kiev - Oslo	Airbus S.A.S	Air Europa	Seventh crew	2020-07-02	15:00:00	20:00:00	05:00:00	Direct flight
Kiev - Ankara	Cessna Aircraft	Air Europa	Nineteenth crew	2020-07-02	11:40:00	14:00:00	02:20:00	Direct flight
Kiev - Copenhagen	Boeing	Air Europa	Fourth crew	2020-07-02	18:30:00	21:00:00	02:30:00	Direct flight
Kiev - Graz	Bombardier Aerospace	Air Europa	Sixteenth crew	2020-07-02	08:20:00	10:40:00	02:20:00	Direct flight
Kiev - Zurich	Cessna Aircraft	Air Europa	Twentieth crew	2020-07-02	18:00:00	20:45:00	02:45:00	Direct flight
Moscow - Kiev	Airbus S.A.S	Delta Air Lines	Fifth crew	2020-07-02	12:00:00	15:00:00	03:00:00	Direct flight
Berlin - Kiev	Bombardier Aerospace	Air Berlin	Thirteenth crew	2020-07-02	16:00:00	18:30:00	02:30:00	Direct flight
Thessaloniki - Kiev	Airbus S.A.S	Air Europa	Sixth crew	2020-07-02	07:00:00	10:00:00	03:00:00	Direct flight
Paris - Kiev	Beechcraft	Air France	Tenth crew	2020-07-02	17:00:00	21:00:00	04:00:00	Direct flight
Barcelona - Kiev	Boeing	Air Europa	Third crew	2020-07-02	14:00:00	18:00:00	04:00:00	Direct flight
Bergen - Kiev	Beechcraft	Air Europa	Eleventh crew	2020-07-02	16:00:00	22:00:00	06:09:00	Direct flight
Istanbul - Kiev	Bombardier Aerospace	Air Europa	Fifteenth crew	2020-07-02	21:00:00	23:00:00	02:00:00	Direct flight
Aarhus - Kiev	Airbus S.A.S	Air Europa	Eighth crew	2020-07-02	12:00:00	15:00:00	03:00:00	Direct flight
Vein - Kiev	Beechcraft	Air Europa	Twelfth crew	2020-07-02	20:20:00	22:20:00	02:00:00	Direct flight

Рисунок 3.6 – розклад польотів для пасажирів

Клієнту надається можливість пошуку по даті, рейсу, та типу польоту (виліт, приліт). За допомогою розкладу пасажир може побачити інформацію про майбутні вильоти та прильоти.

Для користувачів з роллю диспетчер доступний повний розклад польотів, в якому є вся необхідна інформація для контролю вильотів та польотів літаків.

					ДР.122.4142.03.ПЗ.Р03	Аркуш
						80
Зм.	Аркуш	№ документа	Підпис	Дата		

Розклад польотів для диспетчерів представлено на рисунку 3.7.

### Schedule

Flight  Date  Type

Flight name	Source airport name	Destination airport name	Aircraft brand	Aircraft tail number	Crew name	Departure date	Departure time	Arrival time	Flight time	
Charkow - Kiev	Charkow airport	Kiev airport	Boeing	707	First crew	2020-07-01	10:00:00	12:00:00	02:00:00	<input type="button" value="Details"/>
Saint petersburg - Kiev	Saint petersburg airport	Kiev airport	Beechcraft	13	Ninth crew	2020-07-01	16:00:00	20:00:00	04:00:00	<input type="button" value="Details"/>
Cologne - Kiev	Cologne airport	Kiev airport	Boeing	123	Second crew	2020-07-01	18:00:00	21:00:00	03:00:00	<input type="button" value="Details"/>
Athens - Kiev	Athens airport	Kiev airport	Cessna Aircraft	9	Seventeenth crew	2020-07-01	16:00:00	18:40:00	02:40:00	<input type="button" value="Details"/>
Marseilles - Kiev	Marseilles airport	Kiev airport	Bombardier Aerospace	41	Fourteenth crew	2020-07-01	11:30:00	16:00:00	04:30:00	<input type="button" value="Details"/>
Madrid - Kiev	Madrid airport	Kiev airport	Cessna Aircraft	23	Eighteenth crew	2020-07-01	14:00:00	18:40:00	04:40:00	<input type="button" value="Details"/>
Oslo - Kiev	Oslo airport	Kiev airport	Airbus S.A.S	111	Seventh crew	2020-07-01	07:00:00	12:00:00	05:00:00	<input type="button" value="Details"/>
Ankara - Kiev	Ankara airport	Kiev airport	Cessna Aircraft	44	Nineteenth crew	2020-07-01	21:00:00	23:20:00	02:20:00	<input type="button" value="Details"/>
Copenhagen - Kiev	Copenhagen airport	Kiev airport	Boeing	322	Fourth crew	2020-07-01	09:00:00	11:30:00	02:30:00	<input type="button" value="Details"/>
Graz - Kiev	Graz airport	Kiev airport	Bombardier Aerospace	33	Sixteenth crew	2020-07-01	14:40:00	17:00:00	02:20:00	<input type="button" value="Details"/>

Рисунок 3.7 – розклад польотів для диспетчерів

Натиснувши на кнопку Details, яка знаходиться біля польоту в розкладі для диспетчерів, буде перенаправлено на сторінку повної інформації про виліт.

Повна інформація про політ представлена на рисунку 3.8.

### Details

All information about departure

<b>ID:</b>	12
<b>Flight:</b>	Charkow - Kiev; Flight type: Direct flight
<b>Airline:</b>	Air Europa
<b>Source airport:</b>	Charkow airport; Charkow; Ukraine
<b>Destination airport:</b>	Kiev airport; Kiev; Ukraine
<b>Aircraft information:</b>	Aircraft brand: Boeing; Tail number: 707; Num of seats: 200; Fuel: PT; Speed: 700
<b>Crew name:</b>	First crew
<b>Pilot:</b>	First name: Artur; Last name: Boris; Phone number: 0631500843; Position: First pilot; Date of birth: 1998-11-14
<b>Pilot:</b>	First name: Robert; Last name: Alexander; Phone number: 346346346; Position: Fifth pilot; Date of birth: 1981-10-08
<b>Departure time:</b>	10:00:00
<b>Arrival time:</b>	12:00:00
<b>Flight time:</b>	02:00:00
<b>Purchased tickets:</b>	0
<b>Registered passengers:</b>	0
<b>Checked baggage:</b>	0

[Back to List](#)

Рисунок 3.8 – повна інформація про політ



Користувачу з роллю адміністратора надаються можливості адміністрування підсистеми, додавання, вилучення, редагування та перегляду інформації.

Сторінка адміністрування літаків представлена на рисунку 3.9.

AirCraftBrend	Speed	NumberOfSeats	Fuel	TailNum	Edit	Delete
Boeing	700	200	PT	707	Edit	Delete
Boeing	800	150	PT	123	Edit	Delete
Boeing	600	300	PT	703	Edit	Delete
Boeing	900	100	PT	322	Edit	Delete
Airbus S.A.S	800	200	TC-1	231	Edit	Delete
Airbus S.A.S	860	155	TC-1	312	Edit	Delete
Airbus S.A.S	650	324	TC-1	111	Edit	Delete
Airbus S.A.S	700	252	TC-1	125	Edit	Delete
Beechcraft	1000	50	T6	13	Edit	Delete

Рисунок 3.9 – сторінка адміністрування літаків

Для додавання нового літак треба натиснуту на кнопку Create new. Сторінка додавання літака представлена на рисунку 3.10.

Create Aircraft

AirCraftBrend

Speed

NumberOfSeats

Fuel

TailNum

Create

[Back to List;](#)

Рисунок 3.10 – сторінка додавання літака

Для редагування нового літак треба натиснуту на кнопку Edit. Сторінка редагування літака представлена на рисунку 3.11.

Рисунок 3.11 – сторінка редагування літака

Для вилучення літак з підсистеми треба натиснути на кнопку Delete.

Адміністрування інших частин підсистеми здійснюється таким же чином.

Користувачу з роллю працівник з реєстрації, доступні функції реєстрації пасажирів та багажу на виліт.

Сторінки реєстрації пасажира та багажу на виліт представлені на рисунках 3.12 – 3.13.

Рисунок 3.12 – сторінка реєстрації пасажира.

### Baggage check-in

Ticket number	<input type="text" value="2"/>
First name	<input type="text" value="Vova"/>
Last name	<input type="text" value="Naremskiy"/>
Passport series	<input type="text" value="AAAAA"/>
Passport number	<input type="text" value="11111"/>
Baggage weight	<input type="text" value="5,2"/>

Up to 20 kilograms

[Back to home](#)

Рисунок 3.13 – сторінка реєстрації багажу.

У разі успішної реєстрації пасажира, здійснюється перенаправлення на сторінку успішної реєстрації, на якій вказується номер посадкового місця пасажира. Ця сторінка представлена на рисунку 3.14.

### Registration successful

#### Ticket information

First name	Vova
Last name	Naremskiy
Passport series	AAAAA
Pssport number	11111
Place number	147
Class	First class
Flight:	Kiev - Moscow
Departure date:	2020-07-01
Departure time:	08:00:00
Arrival time:	11:00:00
Flight time:	03:00:00
Aircraft information:	Aircraft brand: Airbus S.A.S; Tail number: 231;

[Back to home](#)

Рисунок 3.14 – сторінка успішної реєстрації пасажира.



У разі успішної реєстрації багажу, здійснюється перенаправлення на сторінку успішної реєстрації багажу, на якій вказується номер багажу. Ця сторінка представлена на рисунку 3.15.

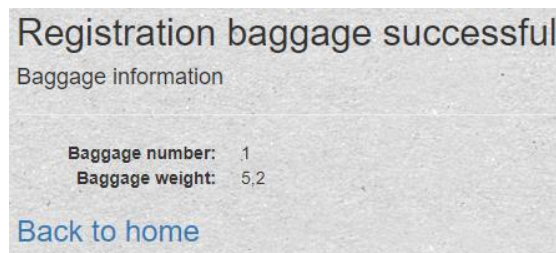


Рисунок 3.15 – сторінка успішної реєстрації багажу.

Користувачу з роллю працівник з посадки, прийому пасажирів та завантажування і розвантажування багажу доступні сторінка по формування списку багажу та пасажирів на виліт. Сторінка формування списків представлена на рисунку 3.16.

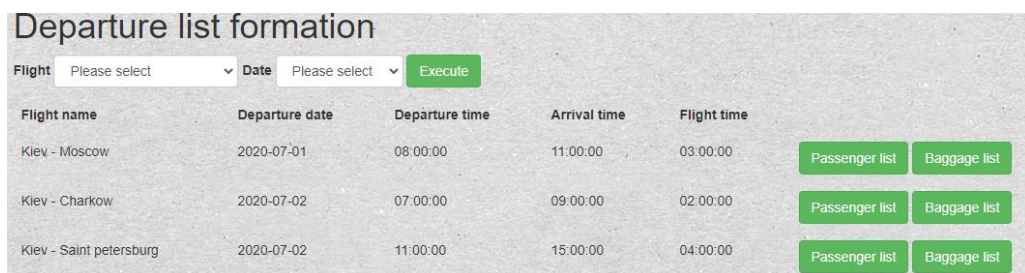


Рисунок 3.16 – сторінка формування списків.

Для формування списку пасажирів на виліт треба натиснути на кнопку Passenger list.

Сторінка списку пасажирів на виліт представлена на рисунку 3.17.

Passenger list										
Ticket number	First name:	Last name:	Passport series	Passport number	Age	Phone	Email	Place number	Ticket price	Class
2	Vova	Naremskiy	AAAAA	111111	21	0635200523	vova@google.com	147	1800	First class
3	Egor	Kamazov	BBBBB	22222	25	0931230531	kamazov@google.com	73	900	Business class

Рисунок 3.17 – список пасажирів на виліт.

Для формування списку багажу на виліт треба натиснути на кнопку Baggage list.

Сторінка списку багажу на виліт представлена на рисунку 3.18.

Baggage list				
Baggage number	Baggage weight	Ticket number	First name	Last name
1	5,2	2	Vova	Naremskiy
2	19,2	3	Egor	Kamazov

Рисунок 3.18 – список багажу на виліт.

Користувачу з роллю економіст доступна сторінка формування цін на авіаквитка. Надається можливість управління аеропортовими зборами та тарифами на виліт. Економіст має можливість додавати, редагувати, вилучати, переглядати аеропортові збори та тарифи на виліт.

У разі виникнення питань по підсистемі існує контактна сторінка, яка представлена на рисунку 3.19.

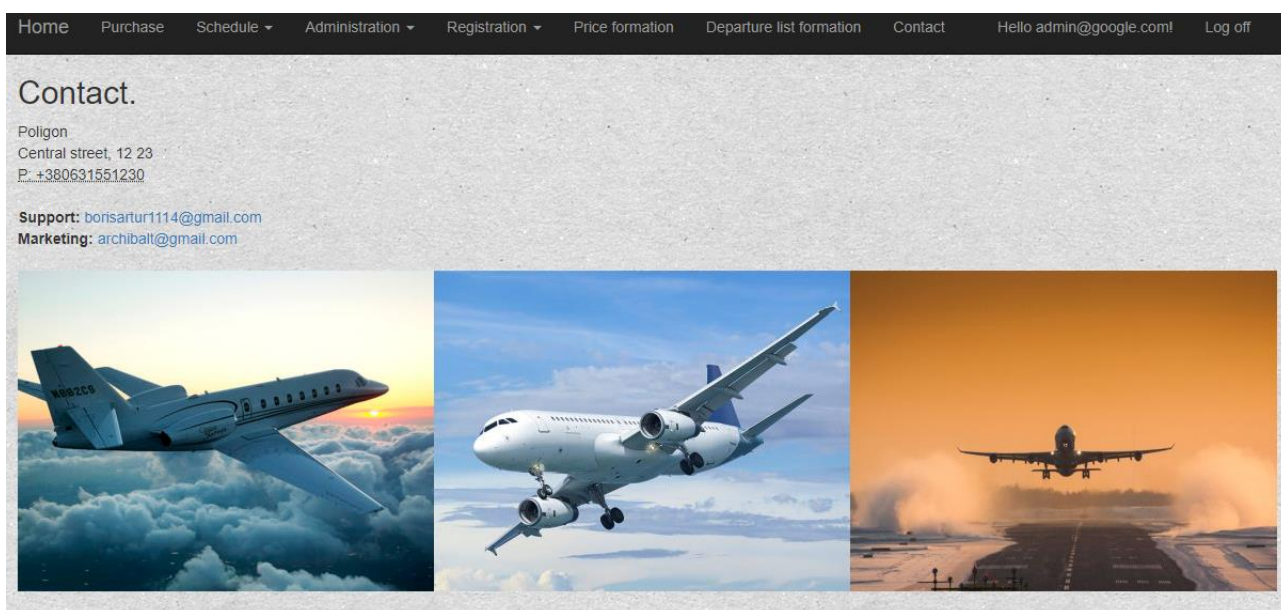


Рисунок 3.19 – контактна сторінка.

### 3.4. Висновки

В результаті реалізації підсистеми “Обслуговування пасажирів” ІС аеропорту, розроблено технічне завдання на розробку, реалізовано програмний продукт за допомогою мови C#, який відповідає всім поставленим раніше вимогам. Розроблено веб-інтерфейс користувача, використовуючи сучасні веб-

технології: HTML5, CSS3, JavaScript, Bootstrap. Розроблений інтерфейс відповідає усім поставленим вимогам і може бути використаний без установки стороннього програмного забезпечення на будь-якому персональному комп'ютері або мобільному пристрої з доступом до мережі інтернет. Також було описано інструкцію з використання підсистеми, яка призначена для ознайомлення користувача з підсистемою.

					ДР.122.4142.03.ПЗ.Р03	Аркуш
						87
Зм.	Аркуш	№ документа	Підпис	Дата		



					ДР.122.4142.03.ПЗ.Р04			
Зм.	Аркаш	№ документа	Підпис	Дата				
					Охорона праці	Літ.	Аркуш	Аркушів
Студент		Борис А. В.					88	12
Керівник		Маршак О. І.				НУК		
Консульт.		Гурець Н. В.						
Зав. каф.		Михелєв І. Л.						

## 4. ОХОРОНА ПРАЦІ

### 4.1. Вступ

Закон України «Про охорону праці» визначає положення про реалізацію конституційного права громадян на охорону їхнього життя й здоров'я в процесі праці, регулює при участі відповідних державних органів відносини між власником підприємства, установи або організації, або вповноваженим їм органом і працівником з питань безпеки, гігієни праці й виробничого середовища й установлює єдиний порядок організації охорони праці в Україні.

*Охорона праці* — це система правових, соціально-економічних, організаційно-технічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження здоров'я і працездатності людини в процесі праці. В поняття охорони праці входять і всі ті заходи, що спеціально призначені для створення особливих полегшених умов праці для жінок і неповнолітніх, а також працівників зі зниженою працездатністю.

*Завдання охорони праці:*

- проектування підприємств, технологічних процесів і конструювання обладнання з обов'язковим виконанням вимог охорони праці;
- знаходження оптимальних співвідношень між різними факторами виробничого середовища, що дозволяє забезпечити мінімум несприятливого впливу їх на здоров'я працівників;
- встановлення, законодавче оформлення визначених норм кожного з несприятливих або небезпечних факторів, систематичний контроль за їх застосуванням;
- розробка конкретних заходів щодо покращення умов праці та забезпечення її безпеки на основі застосування у виробництві новітніх досягнень науки і техніки;
- застосування раціональних засобів захисту працівників від впливу несприятливих факторів виробничого середовища, а також втілення

					ДР.122.4142.03.ПЗ.Р04	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		89

організаційних заходів, які нейтралізують або послаблюють ступінь їх впливу на організм людини;

- розробка та застосування методів і засобів оцінки ефективності заходів з охорони праці, що плануються і здійснюються.

#### **4.2.Аналіз шкідливих та небезпечних факторів на робочому місці**

На робочому місці користувача ПК виникають небезпечні та шкідливі фактори: підвищений рівень шуму, несприятливі мікрокліматичні умови, недостатній рівень освітленості, шкідливі речовини, підвищений рівень електромагнітних випромінювань радіочастот, висока напруга електричної мережі, статична електрика та інші. Робота з ПК супроводжується також підвищеним ступенем напруженості трудового процесу. До хімічно небезпечних факторів, що постійно діють на користувача ПК, відноситься виникнення активних часток у результаті іонізації повітря при роботі комп'ютера. Біологічні шкідливі виробничі фактори в даному приміщенні відсутні. Неправильна організація робочого місця сприяє загальній і локальній напрузі м'язів шиї, тулуба, верхніх кінцівок, скривленню хребта й розвитку остеохондрозу.

Робота програміста пов'язана з значним зоровим навантаженням, що вимагає забезпечення належного освітлення. Інженер-програміст працює з ЕОМ та іншим офісним обладнанням, що є джерелом небезпеки ураження електричним струмом. Трудова діяльність програміста пов'язана з постійним перебуванням в приміщенні, тому для комфортних умов праці необхідно створити належний мікроклімат в комп'ютерній лабораторії. Згідно з нормативними документами [18] та [19] можна виділити такі шкідливі виробничі чинники, що діють на програміста:

- Недостатня освітленість робочого місця;
- Мікроклімат робочої зони: температура, відносна вологості, швидкість руху повітря.

					ДР.122.4142.03.ПЗ.Р04	Аркуш
						90
Зм.	Аркуш	№ документа	Підпис	Дата		

- Підвищений рівень шуму на робочому місці.
- Небезпечна напруга в електричному ланцюзі.
- Підвищений рівень вібрації.

Робота програміста пов'язана з значним зоровим навантаженням, що вимагає забезпечення належного освітлення. В приміщенні рівень природного освітлення і штучного світла повинен відповідати нормам.

Згідно нормативним документам [21] та [22] можна виділити такі шкідливі виробничі чинники, що зазвичай простежуються в робочих приміщеннях:

1. Недостатній рівень штучного освітлення.
2. Мікроклімат робочої зони: температура, відносна вологості, швидкість руху повітря.
3. Підвищений рівень шуму на робочому місці.
4. Небезпечна напруга в електричному ланцюзі.
5. Підвищений рівень вібрації.

#### **4.2.1. Рівень штучного освітлення**

Робота, яка виконується з використанням обчислювальної техніки, мають такі недоліки:

- ймовірність появи прямий блескості;
- погіршена контрастність між зображенням і фоном;
- відображення екрану.

У зв'язку з тим, що природне освітлення слабке, на робочому місці повинно застосовуватися також штучне освітлення.

Основним документом, який регламентує норми освітленості є [20].

Для створення сприятливих умов зорової роботи освітлення робочих приміщень повинно задовольняти таким умовам:

- рівень освітленості робочих поверхонь має відповідати гігієнічним нормам для даного виду роботи

- мають бути забезпечені рівномірність і часова стабільність рівня освітленості у приміщенні, відсутність різких контрастів між освітленістю робочої поверхні та навколишнього простору, відсутність на робочій поверхні різких тіней (особливо рухомих);
- у полі зору предмета не повинно створюватися сліпучого блиску; • штучне світло, що використовується на підприємствах, за своїм спектральним складом має наближатися до природного;
- не створювати небезпечних і шкідливих факторів (шум, теплові випромінювання, небезпеку ураження струмом, пожежо- та вибухонебезпечність);
- бути надійним, простим в експлуатації та економічним.

Категорія виконуваних робіт програміста відноситься до робіт високої точності з присвоєнням розряду III в.

Тому нормативне значення загального освітлення робочого приміщення повинно бути  $E = 300-500 \text{ Лк}$ .

Освітлення на робочому місці програміста повинно бути таким, щоб працівник міг без напруги зору виконувати свою роботу. Розрахунок освітленості робочого місця зводиться до вибору системи освітлення, визначенню необхідного числа світильників, їхнього типу і розміщення. Відповідно до вибраного розрядом зорових робіт допустиме значення освітленості робочої поверхні приймається  $E = 400 \text{ лк}$ .

Для місцевого освітлення робочих місць слід використовувати світильники з відбивачами, що не просвічуються. Світильники повинні розташовуватися так, щоб їх елементи, які світяться, не потрапляли в поле зору працюючих на освітленому робочому місці і на інших робочих місцях.

#### 4.2.2. Мікроклімат робочого зони

Мікроклімат виробничих приміщень - це клімат внутрішнього середовища цих приміщень, який визначається діючими на організм людини поєднаннями температури, вологості і швидкості руху повітря.

					ДР.122.4142.03.ПЗ.Р04	Аркуш
						92
Зм.	Аркуш	№ документа	Підпис	Дата		

Для створення і автоматичної підтримки в лабораторії незалежно від зовнішніх умов оптимальних значень температури, вологості, чистоти і швидкості руху повітря, в холодну пору року використовується водяне опалення, в теплу пору року застосовується кондиціонування повітря. Кондиціонер являє собою вентиляційну установку, яка за допомогою приладів автоматичного регулювання підтримує в приміщенні задані параметри повітряного середовища.

Відповідно до документу [19] праця програміста за важкістю відноситься до легкої фізичної роботи категорії Іа. Основним документом, який регламентує норми мікроклімату робочої зони є [21].

Комп'ютери і офісна техніка є джерелом істотних тепловиділень, що може привести до підвищення температури і зниження відносної вологості в приміщенні. В приміщеннях, де встановлені комп'ютери, повинні дотримуватися певні параметри мікроклімату. В санітарних нормах встановлені величини параметрів мікроклімату, що створюють комфортні умови. Значення параметрів оптимальних та допустимих параметрів мікроклімату згідно з [20] для приміщень, та фактичних параметрів представлено в таблиці 4.1.

Таблиця 4.1 – Оптимальні та допустимі параметри мікроклімату

Період року	Параметр клімату	Значення	
		Оптимальне	Допустиме
Холодний	Температура повітря в приміщенні	21 -23°C	23 - 25°C
	Відносна вологість	40-60%	75%
	Швидкість руху повітря	0,1м/с	до 0,1м/с
Теплий	Температура повітря в приміщенні	22 - 24°C	23 - 25°C
	Відносна вологість	40-60%	75%
	Швидкість руху повітря	0,1 м/с	0,2-0,1м/с

Для забезпечення комфортних умов використовуються як організаційні методи (раціональна організація проведення робіт залежно від пори року і доби,

чергування праці і відпочинку), так і технічні засоби (вентиляція, кондиціонування повітря, опалювальна система).

#### 4.2.3. Шум та вібрації у робочому приміщенні

У приміщеннях з низьким рівнем загального шуму, яким є лабораторія де працює програміст, джерелами шумових перешкод можуть стати вентиляційні установки, кондиціонери або периферійне устаткування для ЕОМ (плотери, принтери та ін.) Тривала дія цих шумів негативно позначаються на емоційному стані персоналу.

Методи вимірювання шуму та допустимі рівні звукового тиску у октавних смугах частот, еквівалентні рівні звуку на робочому місці регламентовані документом [22].

Шум погіршує умови праці здійснюючи шкідливу дію на організм людини. Працюючі в умовах тривалої шумової дії випробовують дратівливість, головні болі, запаморочення, зниження пам'яті, підвищену стомлюваність, пониження апетиту, болі у вухах тощо. Такі порушення в роботі ряду органів і систем організму людини можуть викликати негативні зміни в емоційному стані людини аж до стресових ситуацій. Під впливом шуму знижується концентрація уваги, порушуються фізіологічні функції, з'являється стомленість у зв'язку з підвищеними енергетичними витратами і нервово-психічною напругою, погіршується мовна комутація. Все це знижує працездатність людини і її продуктивність, якість і безпеку праці.

Допустимі еквівалентні рівні звукового тиску представлені в таблиці 4.2

Таблиця 4.1 - Допустимі еквівалентні рівні звукового тиску

Робоче місце		Рівень звуку, дБ А
Приміщення конструкторських бюро, програмістів обчислювальних машин, лабораторій для теоретичних та дослідних робіт		50
Приміщення керування, робочі кімнати		60
Кабіни спостережень і дистанційного керування:		

- без мовного зв'язку	80
- з мовним зв'язком по телефону	65
Постійні робочі місця і робочі зони у виробничих приміщеннях і на території підприємств	80

Для пониження рівня шуму необхідна додаткова звукоізоляція. У якості звукоізолюючих матеріалів, які застосовують у конструкціях перекриттів для зниження передачі структурного (ударного) звуку переважно використовують мати та плити із скляного та мінерального волокна, м'які плити з деревних стружок, картон, гуму, утеплений лінолеум, а також заміна вікон на звукоізолюючі.

В якості заходів щодо зниження шуму можна запропонувати наступне: - -

- облицювання стелі та стін звукопоглинальним матеріалом (знижує шум на 6 - 8 дБ);
- екранування робочого місця (постановкою перегородок, діафрагм);
- установка в комп'ютерних приміщеннях обладнання, що виробляє мінімальний шум;
- раціональна планування приміщення.

#### **4.3. Ергономіка робочого місця**

Виробнича діяльність програміста, змушує його тривалий час перебувати в сидячому положенні, яке є вимушеною позою, тому організм постійно відчуває нестачу в рухливості і активної фізичної діяльності. П

При виконанні роботи сидячи велику роль відіграє плечовий пояс. Переміщення рук у просторі впливає не тільки на роботу м'язів плечового пояса і спини, а й на положення хребта, тазу і навіть ніг.

Щоб виключити виникнення захворювань необхідно мати можливість вільної зміни поз. Необхідно дотримуватися режиму праці і відпочинку з перервами, заповнюваними "відволікаючими" м'язовими навантаженнями на ті ланки опорно-рухового апарату, які не включені в підтримку основної робочої пози.



Антропологічні характеристики людини визначають габаритні і компоновальні параметри його робочого місця, а також вільні параметри окремих його елементів.

За умовами роботи робоче місце програміста відноситься до індивідуального робочого місця для роботи сидячи.

Щоб виключити виникнення захворювань необхідно мати можливість вільної зміни поз. Необхідно дотримуватися режиму праці і відпочинку з перервами, заповнюваними "відволікаючими" м'язовими навантаженнями на ті ланки опорно-рухового апарату, які не включені в підтримку основної робочої пози.

Антропологічні характеристики людини визначають габаритні і компоновальні параметри його робочого місця, а також вільні параметри окремих його елементів.

За умовами роботи робоче місце програміста відноситься до індивідуального робочого місця для роботи сидячи.

Проектування робочих місць, забезпечених відео терміналами, відноситься до числа важливих проблем ергономічного проектування в області обчислювальної техніки.

Робоче місце і взаємне розташує всіх його елементів повинне відповідати антропометричним, фізичним і психологічним вимогам. Велике значення має також характер роботи. Зокрема, при організації робочого місця програміста повинні бути дотримані наступні основні умови: оптимальне розміщення устаткування, що входить до складу робочого місця і достатній робочий простір, що дозволяє здійснювати всі необхідні рухи і переміщення.

Ергономічними аспектами проектування відео термінальних робочих місць, зокрема, є: висота робочої поверхні, розміри простору для ніг, вимоги до того, що розташовує документів на робочому місці (наявність і розміри підставки для документів, можливість різного розміщення документів, відстань від очей користувача до екрану, документа, клавіатури тощо), характеристики робочого крісла, вимоги до поверхні робочого столу,

					ДР.122.4142.03.ПЗ.Р04	Аркуш
						96
Зм.	Аркуш	№ документа	Підпис	Дата		

можливість регулювання елементів робочого місця . Головними елементами робочого місця програміста є стіл і крісло. Основним робочим положенням є положення сидячи.

Робоча поза сидячи викликає мінімальне стомлення програміста. Рациональне планування робочого місця передбачає чіткий порядок і постійність розміщення предметів, засобів праці і документації. Те, що потрібне для виконання робіт частіше, розташоване в зоні легкої досяжності робочого простору.

Моторне поле – простір робочого місця, в якому можуть здійснюватися рухові дії людини.

Максимальна зона досяжності рук - це частина моторного поля робочого місця, обмеженого дугами, описуваними максимально витягнутими руками при русі їх в плечовому суглобі.

Оптимальна зона - частина моторного поля робочого місця, обмеженого дугами, описуваними передпліччями при русі в ліктьових суглобах з опорою в точці ліктя і з відносно нерухомим плечем. Схематичне зображення зони досягнення рук представлено на рисунку 4.1.

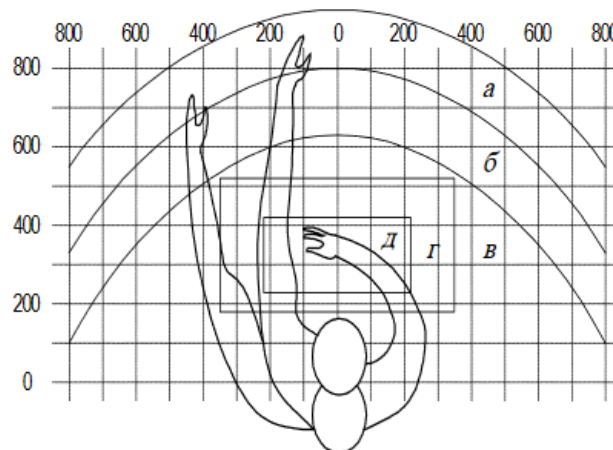


Рисунок 4.1 – Зони досяжності рук в горизонтальній площині:

- а) – зона максимальної досяжності;
- б) – зона досяжності пальців при витягнутій руці;
- в) – зона легкої досяжності долоні;
- г) – оптимальний простір для грубої ручної роботи;

е) – оптимальний простір для тонкої ручної роботи.

*Оптимальне розміщення предметів праці і документації в зонах досяжності:*

- 1.Дисплей розміщується в зоні **а** (в центрі).
- 2.Системний блок розміщується в передбаченій ніші столу.
- 3.Клавіатура – в зоні **г/д**.
- 4.«миша» – в зоні **в** справа.
- 5.Сканер в зоні **а/б**(зліва).
- 6.Принтер знаходиться в зоні **а**(справа).
- 7.Документація: необхідна при роботі – в зоні легкої досяжності долоні – **в**, а у висувних ящиках столу – література, невживана постійно.

На рисунку 4.2 показаний приклад розміщення основних і периферійних складових ПК на робочому столі програміста.

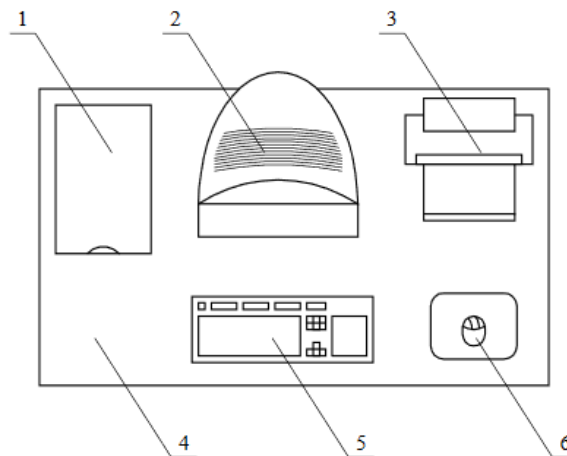


Рисунок 4.2 – Розміщення основних і периферійних складових ПК: 1 – сканер, 2 – монітор, 3 – принтер, 4 – поверхня робочого столу, 5 – клавіатура, 6 – маніпулятор типу «миша»

*Для комфортної роботи стіл повинен задовольняти наступним умовам :*

1. висота столу повинна бути вибрана з урахуванням можливості сидіти вільно, в зручній позі, при необхідності спираючись на підлокітники;

2. нижня частина столу повинна бути сконструйована так, щоб програміст міг зручно сидіти, не був вимушений підтискати ноги;
3. поверхня столу повинна володіти властивостями, що виключають появу відблисків в полі зору програміста;
4. конструкція столу повинна передбачати наявність висувних ящиків (не менше 3 для зберігання документації, лістингів, канцелярських обладнань);
5. висота робочої поверхні рекомендується в межах 680-760 мм. Висота поверхні, на яку встановлюється клавіатура, повинна бути біля 650 мм.

Велике значення надається характеристикам робочого крісла. Так, висота сидіння над рівнем підлоги, що рекомендується, знаходиться в межах 420-550 мм. Поверхня сидіння м'яка, передній край закруглює, а кут нахилу спинки - регульований.

Необхідно передбачати при проектуванні можливість різного розміщення документів: збоку від відео-терміналу, між монітором і клавіатурою тощо. Крім того, у випадках, коли відео-термінал має низьку якість зображення, наприклад помітні мигтіння, відстань від очей до екрану роблять більше (біля 700мм), ніж відстань від ока до документа (300-450мм). Взагалі при високій якості зображення на відео-терміналі відстань від очей користувача до екрану, документа і клавіатури може бути рівним.

Робоче місце програміста повинне займати площу не менше 6 м, висота приміщення повинна бути не менше 4 м, а обсяг - не менше 20 м<sup>3</sup> на одну людину.

## ВИСНОВКИ

В результаті виконання дипломної роботи розроблено підсистему “Обслуговування пасажирів” ІС аеропорту.

У розділі 1 описано та проаналізовано об’єкт автоматизації та його організаційну структуру. Розроблено функціональні, програмні та технічні вимоги до підсистеми, що дозволяють якісно поліпшити роботу підприємства. Визначено ціль та задачі створення інформаційної підсистеми. Розроблено концепцію, функціональну модель блок-схему підсистеми, яка спрямована на покращення робочого процесу працівників відділу підприємства, що призведе до підвищення ефективності роботи.

У розділі 2 було розроблено технічний проект підсистеми з детальним описом вимог та засобами їх досягнення. Розроблено діаграми використання та діяльності, концептуальну, логічну, фізичну модель бази даних. Описано структуру програмного забезпечення та процес роботи з підсистемою.

У розділі 3 було реалізовано підсистему “Обслуговування пасажирів” ІС аеропорту, розроблено технічне завдання на розробку, реалізовано програмний продукт, який відповідає всім поставленим раніше вимогам. Розроблено веб-інтерфейс користувача. Також було описано інструкцію з використання підсистеми, яка призначена для ознайомлення користувача з підсистемою.

Крім того у розділі 4 було проведено аналіз шкідливих та небезпечних факторів на робочому місці програміста, описано необхідний рівень штучного освітлення для комфортної роботи, мікроклімат робочої зони, шум та вібрації в приміщенні, ергономіку робочого місця необхідну для мінімізації шкоди здоров’ю програміста.

Реалізація та впровадження підсистеми збільшить якість обслуговування пасажирів, конкурентоспроможність, популярність, рентабельність та працездатність відділу обслуговування пасажирів.

					ДР.122.4142.03.ПЗ	Аркуш
						100
Зм.	Аркуш	№ документа	Підпис	Дата		

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. А.В. Маслов. “Проектування інформаційних систем в економіці”. Навчальний посібник/ А.В. Маслов – Томськ. Видавництво Томського політехнічного університету, 2008. – 216 с.
2. Міжнародний стандарт ISO/IEC 12201-2010 Інформаційна технологія. Системна та програмна інженерія. Процеси життєвого циклу програмних засобів.
3. Методичні вказівки до виконання дипломного проектування зі спеціальності 6.050101 “Комп'ютерні науки” кваліфікації “Бакалавр інформаційних управляючих систем і технологій” / В. Ф. Ажищев, К. В. Кошкін, Г. С. Морозова. – Миколаїв : НУК, 2014. – 57 с.
4. ГОСТ 19.003-80. “Схеми алгоритмів та програм. Позначення умовні графічні.”
5. А.В. Вендров “CASE-технології. Сучасні методи і засоби проектування інформаційних систем”. с. 123.
6. Ю.Н. Алпатов. “Моделювання процесів та систем управління”. Навчальний посібник. – СПб.: Видавництво “Лань”, 2018. – 140с. (Посібники для вузів. Спеціальна література).
7. Хоменко А.Д., Циганков В.М., Мальцев М.Г. “Бази даних: Навчальний посібник для вищих навчальних закладів”. Під редакцією професора А.Д. Хоменка – 6-е видання. доп. – СПб.: КОРОНА-Вік. 2009 – 736с.
8. ЗВІТ ПРО УПРАВЛІННЯ Державне підприємство «Міжнародний Аеропорт «Бориспіль» [Електронний ресурс] - <https://kbp.aero/>. Режим доступу: <https://kbp.aero/wp-content/uploads/2019/05/Zvit-proupravlinnya-DPMA-Boryspil-2018.pdf>. – Дата доступу: 21.04.2020.
9. Міжнародний Аеропорт Бориспіль [Електронний ресурс] -<https://lb.ua/> Режим доступу: [https://lb.ua/file/company/3436\\_borispil\\_aeroport.html](https://lb.ua/file/company/3436_borispil_aeroport.html). – Дата доступу: 22.04.2020.

					ДР.122.4142.03.ПЗ	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		101

10. Схема терміналів А, В, С, F аеропорту Бориспіль [Електронний ресурс] - <http://aviabilet-online.com/>. Режим доступу: <http://aviabilet-online.com/Kiev/schema.asp>. – Дата доступу: 23.04.2020.
11. Системы бронирования авиабилетов [Електронний ресурс] - <https://flyworld.ru/>. Режим доступу: <https://flyworld.ru/stati/sistemy-bronirovaniya-aviabiletov/>. – Дата доступу: 01.06.2020.
12. Amadeus [Електронний ресурс] - <http://support.nemo.travel/ru>. Режим доступу: <http://support.nemo.travel/ru/Amadeus>. – Дата доступу: 02.06.2020.
13. Travelport [Електронний ресурс] - <http://support.nemo.travel/ru/>. Режим доступу: <http://support.nemo.travel/ru/Travelport>. – Дата доступу: 03.06.2020.
14. Авиабилеты. IT системы бронирования [Електронний ресурс] - <https://habr.com/ru>. Режим доступу: <https://habr.com/ru/company/buruki/blog/192384/>. – Дата доступу: 03.06.2020.
15. Metanit // C#/.NET [Електронний ресурс] - <https://metanit.com/>. Режим доступу: <https://metanit.com/sharp/tutorial/1.1.php>. – Дата доступу: 02.06.2020.
16. MVC модель-представление-контроллер [Електронний ресурс] - <https://web-creator.ru/>. Режим доступу: <https://web-creator.ru/articles/mvc>. – Дата доступу: 27.05.2020.
17. Діаграма класів [Електронний ресурс] - <http://mmsa.kpi.ua/>. Режим доступу: [http://mmsa.kpi.ua/sites/default/files/disciplines/%D0%A0%D0%BE%D0%B7%D1%80%D0%BE%D0%B1%D0%BA%D0%B0%20%D1%96%20%D1%82%D0%B5%D1%81%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC/didkovska\\_m\\_v\\_testing\\_lecture\\_5.pdf](http://mmsa.kpi.ua/sites/default/files/disciplines/%D0%A0%D0%BE%D0%B7%D1%80%D0%BE%D0%B1%D0%BA%D0%B0%20%D1%96%20%D1%82%D0%B5%D1%81%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC/didkovska_m_v_testing_lecture_5.pdf). – Дата доступу: 27.05.2020.

18. ССБТ «Небезпечні і шкідливі виробничі фактори. Класифікація». ДСТУ 12.0.003-74\*. [Електронний ресурс]: ДСТУ 12.0.003-74\*. – М., 1980 – Режим доступу: [http://online.budstandart.com/ua/catalog/doc-page.html?id\\_doc=48127](http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=48127). – Дата доступу: 21.05.2020.
19. «Гігієнічна класифікація праці за показниками шкідливості та небезпечності факторів виробничого середовища». [Електронний ресурс] zakon.rada.gov.ua/laws: – Режим доступу: <https://zakon.rada.gov.ua/laws/show/z0472-14>. – Дата доступу : 21.05.2020.
20. «Державні будівельні норми України. Природне та штучне освітлення». ДБН В .2.5-28:2018. [Електронний ресурс]: ДБН В .2.5-28:2018. – Режим доступу: [https://ledeffect.com.ua/images/\\_branding/dbn2018.pdf](https://ledeffect.com.ua/images/_branding/dbn2018.pdf) . – Дата доступу: 22.05.2020.
21. «Державні санітарні норми мікроклімату виробничих приміщень». ДСН 3.3.6.042-99. [Електронний ресурс]: ДСН 3.3.6.042-99. – Режим доступу: <http://mozdocs.kiev.ua/view.php?id=1972> . – Дата доступу: 22.05.2020.
22. «Санітарні норми виробничого шуму, ультразвуку та інфразвуку». ДСН 3.3.6.037-99. [Електронний ресурс]: ДСН 3.3.6.037-99. – Режим доступу: <http://normativ.com.ua/types/tdoc4878.php> – Дата доступу : 23.05.2020.



					ДР.122.4142.03.ПЗ			
Зм.	Аркаш	№ документа	Підпис	Дата				
					Додатки	Літ.	Аркуш	Аркушів
Студент		Борис А. В.					104	15
Керівник		Маршак О. І.				НУК		
Зав. каф.		Михелєв І. Л.						

## **Додаток А. Технічне завдання на розробку**

### **1. Вступ**

1.1. Назва – розробка підсистеми “Обслуговування пасажирів” ІС аеропорту.

1.2. Призначення підсистеми – полягає в покращенні якості обслуговування пасажирів, автоматизації процесів обробки даних, збільшенні конкурентоспроможності підприємства та продуктивності роботи відділу з обслуговування пасажирів.

1.3. Область застосування – підприємство “Аеропорт”, відділ обслуговування пасажирів аеропорту.

### **2. Призначення розробки**

Підсистема повинна забезпечити якісний продаж авіаквитків клієнтам аеропорту, побудову та перегляд розкладу вильотів та прильотів для потенційних пасажирів і диспетчерської служби, реєстрацію пасажирів та багажу. Також повинна надавати можливість адміністрування бази даних підсистеми, формування списків пасажирів та багажу на виліт.

### **3. Вимоги до підсистеми**

#### **3.1. Вимоги до функціональних характеристик**

Підсистема міститиме наступні функціональні характеристики:

- Можливість реєстрації;
- Можливість авторизації;
- Можливість управління правами доступу до підсистеми;
- Можливість покупки квитка на необхідний рейс та дату;
- Можливість клієнтам та пасажирам переглядати розклад вильотів та прильотів;
- Можливість диспетчерам переглядати, сортувати, редагувати повний розклад польотів;
- Адміністрування польотів - можливість додавання, вилучення, редагування, перегляду польотів;

					ДР.122.4142.03.ПЗ	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		105

- Адміністрування рейсів - можливість перегляду, додавання, вилучення, редагування рейсів та пунктів відправлення, призначення;
- Адміністрування літаків - можливість додавання, вилучення, редагування, перегляд літаків;
- Адміністрування екіпажів - можливість додавання, вилучення, редагування, перегляду екіпажів, додавання пілотів в екіпаж та вилучення;
- Адміністрування пілотів - можливість додавання, вилучення, редагування, перегляду пілотів та посад пілотів;
- Реєстрація багажу – додавання, вилучення багажу;
- Процес реєстрації пасажирів – порівняння персональних даних пасажирів з даними в квитку;
- Формування та перегляд списків зареєстрованих пасажирів на виліт;
- Формування та перегляд списків зареєстрованого багажу на виліт;
- Формування ціни на квитки – можливість додавання, вилучення, редагування, перегляду тарифів та аеропортових зборів на виліт, які в сумі складають ціну.

### 3.2. Вимоги до надійності

#### 3.2.1. Вимоги до забезпечення надійності функціонування підсистеми

Надійне (стійке) функціонування підсистеми має бути забезпечено надійним (стійким) функціонуванням мережі інтернет і мережі електропостачання в місці, де буде розміщений сервер.

3.2.2. Відмова виконання програмного продукту через некоректні дії користувача.

- Валідація вхідних даних для коректної роботи.
- Функція автоматичного резервного копіювання кожен тиждень.

### 3.3. Умови експлуатації

Данною інформаційною підсистемою будуть користуватися клієнти (пасажирів), працівників аеропорту (диспетчери, адміністратори, касири та інші

					ДР.122.4142.03.ПЗ	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		106

співпрацівники відділу), та не потребують додаткового навчання по даному ПО, але можуть вимагати додаткову допомогу у роботі з вихідними даними. Окрім того буде надано інструкцію з використання програми.

#### 3.4. Вимоги до інформаційної та програмної сумісності

##### 3.4.1. Вимоги до інформаційних структур і методів розв'язання

Вимоги до інформаційних структур і методів розв'язання відсутні.

##### 3.4.2. Вимоги до вихідних кодів і мов програмування

Додаткові вимоги не пред'являються.

##### 3.4.3. Вимоги до програмних засобів, які використовуються програмою

Програмні засоби що використовуються мають бути вільно поширювані

#### 3.5. Вимоги до маркування та упаковки

##### 3.5.1. Вимоги до маркування :

- Читабельність тексту, чіткість ілюстрацій;
- Достовірність інформації про підсистему

##### 3.5.2. Вимоги до упаковки:

Вимоги до упаковки відсутні.

#### 3.6. Вимоги до транспортування та зберігання

Вимоги до транспортування програми відсутня.

#### 3.7. Спеціальні вимоги

Спеціальні вимоги відсутні.

### 4. Вимоги до програмної документації

#### 4.1. Попередній склад програмної документації

Попередній склад програмної документації повинен включати в себе:

- технічне завдання;
- устав проекту.

#### 4.2. Склад програмної документації

Програмна документація складається з наступних програмних документів:

- Устав проекту;

- Технічне завдання ;
- Опис підсистеми ;
- Код програми ;
- Керівництво користувача.

Сертифікація містить список всіх програмних документів.

Технічне завдання містить опис завдання роботи, склад функціональних характеристик програми, опис вимог для розробки.

Опис програми містить опис логічної структури програми, форматів вхідних та вихідних даних, використовуваних технічних засобів.

Код програми містить опис блоків програми і запис блоків в програмному коді.

Інструкція з використання підсистеми містить інформацію, необхідну для ознайомлення користувача з програмним продуктом.

## **5. Техніко-економічні показники**

### **5.1. Економічні переваги розробки**

Орієнтовна економічна ефективність не розраховується.

## **6. Стадії та етапи розробки програмного забезпечення**

### **6.1. Стадії розробки**

Розробка повинна бути проведена в 5 стадій:

1. Технічне завдання
2. Ескізний проект
3. Технічний проект
4. Робочий проект
5. Впровадження.

## Додаток Б. Скрипт створення бази даних підсистеми

```
CreateTable(
    "dbo.Aircraft",
    c => new
    {
        Id = c.Int(nullable: false, identity: true),
        AirCraftBrend = c.String(nullable: false),
        Speed = c.Int(nullable: false),
        NumberOfSeats = c.Int(nullable: false),
        Fuel = c.String(),
        TailNum = c.Int(nullable: false),
    })
.PrimaryKey(t => t.Id);

CreateTable(
    "dbo.Departures",
    c => new
    {
        Id = c.Int(nullable: false, identity: true),
        CrewId = c.Int(nullable: false),
        FlightId = c.Int(nullable: false),
        AircraftId = c.Int(nullable: false),
        RateId = c.Int(nullable: false),
        AirportTaxId = c.Int(nullable: false),
        DepartureDate = c.DateTime(nullable: false, storeType:
"date"),
        NumOfPurchasedTickets = c.Int(nullable: false),
        NumOfRegisteredPass = c.Int(nullable: false),
        NumOfCheckedBaggage = c.Int(nullable: false),
    })
.PrimaryKey(t => t.Id)
.ForeignKey("dbo.Aircraft", t => t.AircraftId, cascadeDelete:
true)
.ForeignKey("dbo.AirportTaxes", t => t.AirportTaxId,
cascadeDelete: true)
.ForeignKey("dbo.Crews", t => t.CrewId, cascadeDelete: true)
.ForeignKey("dbo.Flights", t => t.FlightId, cascadeDelete: true)
.ForeignKey("dbo.Rates", t => t.RateId, cascadeDelete: true)
.Index(t => t.CrewId)
.Index(t => t.FlightId)
.Index(t => t.AircraftId)
.Index(t => t.RateId)
.Index(t => t.AirportTaxId);

CreateTable(
    "dbo.AirportTaxes",
    c => new
    {
        Id = c.Int(nullable: false, identity: true),
        AirportTaxName = c.String(nullable: false),
        amountOfMoney = c.Int(nullable: false),
    })
.PrimaryKey(t => t.Id);

CreateTable(
    "dbo.Crews",
    c => new
    {
        Id = c.Int(nullable: false, identity: true),
        CrewName = c.String(nullable: false),
        NumberPilotsInCrew = c.Int(nullable: false),
        CrewExperience = c.Int(nullable: false),
```

					ДР.122.4142.03.ПЗ	Аркуш
Зм.	Аркуш	№ документа	Підпис	Дата		109

```

        })
        .PrimaryKey(t => t.Id);

CreateTable(
    "dbo.Pilots",
    c => new
    {
        Id = c.Int(nullable: false, identity: true),
        PositionId = c.Int(),
        CrewId = c.Int(),
        FirstName = c.String(nullable: false),
        LastName = c.String(nullable: false),
        DateOfBirth = c.DateTime(nullable: false, storeType:
"date"),
        Address = c.String(),
        PhoneNum = c.String(nullable: false),
    })
    .PrimaryKey(t => t.Id)
    .ForeignKey("dbo.Crews", t => t.CrewId)
    .ForeignKey("dbo.Positions", t => t.PositionId)
    .Index(t => t.PositionId)
    .Index(t => t.CrewId);

CreateTable(
    "dbo.Positions",
    c => new
    {
        Id = c.Int(nullable: false, identity: true),
        PositionName = c.String(nullable: false),
        Salary = c.Double(nullable: false),
        Prize = c.Double(nullable: false),
        Property = c.Int(nullable: false),
    })
    .PrimaryKey(t => t.Id);

CreateTable(
    "dbo.Flights",
    c => new
    {
        Id = c.Int(nullable: false, identity: true),
        AirlineId = c.Int(nullable: false),
        SourceAirportId = c.Int(nullable: false),
        DestinationAirportId = c.Int(nullable: false),
        FlightName = c.String(nullable: false),
        DepartureTime = c.Time(nullable: false, precision: 7),
        ArrivalTime = c.Time(nullable: false, precision: 7),
        FlightTime = c.Time(nullable: false, precision: 7),
        FlightType = c.String(),
    })
    .PrimaryKey(t => t.Id)
    .ForeignKey("dbo.Airlines", t => t.AirlineId, cascadeDelete:
true)
    .ForeignKey("dbo.Airports", t => t.DestinationAirportId,
cascadeDelete: true)
    .ForeignKey("dbo.Airports", t => t.SourceAirportId)
    .Index(t => t.AirlineId)
    .Index(t => t.SourceAirportId)
    .Index(t => t.DestinationAirportId);

CreateTable(
    "dbo.Airlines",
    c => new
    {

```

					ДР.122.4142.03.ПЗ	Аркуш
						110
Зм.	Аркуш	№ документа	Підпис	Дата		

```

        Id = c.Int(nullable: false, identity: true),
        AirlineName = c.String(nullable: false),
    })
    .PrimaryKey(t => t.Id);

CreateTable(
    "dbo.Airports",
    c => new
    {
        Id = c.Int(nullable: false, identity: true),
        AirportName = c.String(nullable: false),
        CityId = c.Int(nullable: false),
    })
    .PrimaryKey(t => t.Id)
    .ForeignKey("dbo.Cities", t => t.CityId, cascadeDelete: true)
    .Index(t => t.CityId);

CreateTable(
    "dbo.Cities",
    c => new
    {
        Id = c.Int(nullable: false, identity: true),
        NameCity = c.String(nullable: false),
        CountryId = c.Int(nullable: false),
    })
    .PrimaryKey(t => t.Id)
    .ForeignKey("dbo.Countries", t => t.CountryId, cascadeDelete:
true)
    .Index(t => t.CountryId);

CreateTable(
    "dbo.Countries",
    c => new
    {
        Id = c.Int(nullable: false, identity: true),
        Name = c.String(nullable: false),
    })
    .PrimaryKey(t => t.Id);

CreateTable(
    "dbo.Rates",
    c => new
    {
        Id = c.Int(nullable: false, identity: true),
        RateName = c.String(nullable: false),
        amountOfMoney = c.Int(nullable: false),
    })
    .PrimaryKey(t => t.Id);

CreateTable(
    "dbo.TicketSales",
    c => new
    {
        Id = c.Int(nullable: false, identity: true),
        DepartureId = c.Int(nullable: false),
        FirstName = c.String(nullable: false),
        LastName = c.String(nullable: false),
        PassportSeries = c.String(nullable: false),
        PassportNum = c.String(nullable: false),
        Age = c.String(nullable: false),
        PhoneNum = c.String(),
        Email = c.String(),
        PlaceNumber = c.Int(),
    })

```

					ДР.122.4142.03.ПЗ	Аркуш
						111
Зм.	Аркуш	№ документа	Підпис	Дата		



```

        TicketPrice = c.Double(nullable: false),
        Class = c.String(nullable: false),
        Registered = c.Boolean(nullable: false),
    })
    .PrimaryKey(t => t.Id)
    .ForeignKey("dbo.Departures", t => t.DepartureId, cascadeDelete:
true)

    .Index(t => t.DepartureId);

CreateTable(
    "dbo.Baggages",
    c => new
    {
        Id = c.Int(nullable: false, identity: true),
        TicketSaleId = c.Int(nullable: false),
        Weight = c.Double(nullable: false),
    })
    .PrimaryKey(t => t.Id)
    .ForeignKey("dbo.TicketSales", t => t.TicketSaleId,
cascadeDelete: true)
    .Index(t => t.TicketSaleId);

CreateTable(
    "dbo.AspNetRoles",
    c => new
    {
        Id = c.String(nullable: false, maxLength: 128),
        Name = c.String(nullable: false, maxLength: 256),
    })
    .PrimaryKey(t => t.Id)
    .Index(t => t.Name, unique: true, name: "RoleNameIndex");

CreateTable(
    "dbo.AspNetUserRoles",
    c => new
    {
        UserId = c.String(nullable: false, maxLength: 128),
        RoleId = c.String(nullable: false, maxLength: 128),
    })
    .PrimaryKey(t => new { t.UserId, t.RoleId })
    .ForeignKey("dbo.AspNetRoles", t => t.RoleId, cascadeDelete:
true)

    .ForeignKey("dbo.AspNetUsers", t => t.UserId, cascadeDelete:
true)

    .Index(t => t.UserId)
    .Index(t => t.RoleId);

CreateTable(
    "dbo.AspNetUsers",
    c => new
    {
        Id = c.String(nullable: false, maxLength: 128),
        Email = c.String(maxLength: 256),
        EmailConfirmed = c.Boolean(nullable: false),
        PasswordHash = c.String(),
        SecurityStamp = c.String(),
        PhoneNumber = c.String(),
        PhoneNumberConfirmed = c.Boolean(nullable: false),
        TwoFactorEnabled = c.Boolean(nullable: false),
        LockoutEndDateUtc = c.DateTime(),
        LockoutEnabled = c.Boolean(nullable: false),
        AccessFailedCount = c.Int(nullable: false),
        UserName = c.String(nullable: false, maxLength: 256),
    })

```

					ДР.122.4142.03.ПЗ	Аркуш
						112
Зм.	Аркуш	№ документа	Підпис	Дата		

```

        })
        .PrimaryKey(t => t.Id)
        .Index(t => t.UserName, unique: true, name: "UserNameIndex");

CreateTable(
    "dbo.AspNetUserClaims",
    c => new
    {
        Id = c.Int(nullable: false, identity: true),
        UserId = c.String(nullable: false, maxLength: 128),
        ClaimType = c.String(),
        ClaimValue = c.String(),
    })
    .PrimaryKey(t => t.Id)
    .ForeignKey("dbo.AspNetUsers", t => t.UserId, cascadeDelete:
true)
    .Index(t => t.UserId);

CreateTable(
    "dbo.AspNetUserLogins",
    c => new
    {
        LoginProvider = c.String(nullable: false, maxLength:
128),
        ProviderKey = c.String(nullable: false, maxLength: 128),
        UserId = c.String(nullable: false, maxLength: 128),
    })
    .PrimaryKey(t => new { t.LoginProvider, t.ProviderKey, t.UserId
})
    .ForeignKey("dbo.AspNetUsers", t => t.UserId, cascadeDelete:
true)
    .Index(t => t.UserId);

```

## Додаток В. Лістинги модулів

Оскільки підсистема складається з багатьох модулів, нижче наведено тексти деяких (не усіх) програмних модулів.

### Лістинг 1 – AircraftController.cs

```
using Airport_IS.Models;
using Airport_IS.Models.Entities;
using Airport_IS.Models.ViewModels;
using System;
using System.Collections.Generic;
using System.Data.Entity;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace Airport_IS.Controllerssupport
{
    public class AircraftController : Controller
    {
        // GET: Aircraft
        ApplicationDbContext db = new ApplicationDbContext();

        [Authorize(Roles = "admin")]
        public ActionResult AircraftList()
        {
            var airCRAFT = db.Aircrafts;
            return View(airCRAFT);
        }

        [Authorize(Roles = "admin")]
        [HttpGet]
        public ActionResult Create()
        {
            return View();
        }

        [Authorize(Roles = "admin")]
        [HttpPost]
        public ActionResult Create(AircraftView model)
        {
            if (ModelState.IsValid)
            {
                Aircraft aircraft = new Aircraft();
                aircraft.AirCRAFTBrend = model.AirCRAFTBrend;
                aircraft.NumberOfSeats = model.NumberOfSeats;
                aircraft.Speed = model.Speed;
                aircraft.TailNum = model.TailNum;
                aircraft.Fuel = model.Fuel;
                db.Aircrafts.Add(aircraft);
                db.SaveChanges();
                return RedirectToAction("AircraftList", "Aircraft");
            }
            return View();
        }

        [Authorize(Roles = "admin")]
        [HttpGet]
        public ActionResult Edit(int id)
        {
            var aircraft = db.Aircrafts.Find(id);
```

					ДР.122.4142.03.ПЗ	Аркуш
						114
Зм.	Аркуш	№ документа	Підпис	Дата		

```

        return View(aircraft);
    }
    [Authorize(Roles = "admin")]
    [HttpPost]
    public ActionResult Edit(Aircraft model)
    {
        if (ModelState.IsValid)
        {
            Aircraft aircraft = db.Aircrafts.Find(model.Id); ;
            aircraft.AirCraftBrend = model.AirCraftBrend;
            aircraft.NumberOfSeats = model.NumberOfSeats;
            aircraft.Speed = model.Speed;
            aircraft.TailNum = model.TailNum;
            aircraft.Fuel = model.Fuel;
            db.Entry(aircraft).State = EntityState.Modified;
            db.SaveChanges();
            return RedirectToAction("AircraftList", "Aircraft");
        }
        return View();
    }
    [Authorize(Roles = "admin")]
    [HttpGet]
    public ActionResult Delete(int id)
    {
        var aircraft = db.Aircrafts.Find(id);
        if(aircraft == null)
        {
            return View("Error");
        }
        db.Aircrafts.Remove(aircraft);
        db.SaveChanges();
        return RedirectToAction("AircraftList", "Aircraft");
    }
}
}
}

```

## Лістинг 2 – TicketsController.cs

```

using Airport_IS.Models;
using Airport_IS.Models.Entities;
using Airport_IS.Models.ViewModels;
using System;
using System.Collections.Generic;
using System.Globalization;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Data.Entity;

namespace Airport_IS.Controllers
{
    public class TicketsController : Controller
    {
        ApplicationDbContext db = new ApplicationDbContext();

        [HttpGet]
        public ActionResult Purchase()
        {
            SelectList flights = new
            SelectList(db.Flights.Where(p=>p.SourceAirport.AirportName == "Kiev airport"),
            "FlightName", "FlightName");

```

					ДР.122.4142.03.ПЗ	Аркуш
						115
Зм.	Аркуш	№ документа	Підпис	Дата		

```

        List<Departure> departures = db.Departures.ToList();

        List<string> dates = new List<string>();
        foreach (Departure d in departures)
        {
            if (!dates.Contains(d.DepartureDate.ToString("d",
CultureInfo.CreateSpecificCulture("es-ES"))))
            {
                dates.Add(d.DepartureDate.ToString("d",
CultureInfo.CreateSpecificCulture("es-ES")));
            }
        }
        List<string> clss = new List<string>();
        clss.Add("First class");
        clss.Add("Business class");
        clss.Add("Economy class");
        ViewBag.Flights = flights;
        ViewBag.Dates = new SelectList(dates);
        ViewBag.Class = new SelectList(clss);
        return View();
    }

    [HttpPost]
    public ActionResult Purchase(ViewPurchase model, String
FlightName, String Date)
    {
        SelectList flights = new SelectList(db.Flights.Where(p =>
p.SourceAirport.AirportName == "Kiev airport"), "FlightName", "FlightName");
        List<Departure> departures = db.Departures.ToList();
        List<string> dates = new List<string>();
        foreach (Departure p in departures)
        {
            if (!dates.Contains(p.DepartureDate.ToString("d",
CultureInfo.CreateSpecificCulture("es-ES"))))
            {
                dates.Add(p.DepartureDate.ToString("d",
CultureInfo.CreateSpecificCulture("es-ES")));
            }
        }
        List<string> clss = new List<string>();
        clss.Add("First class");
        clss.Add("Business class");
        clss.Add("Economy class");
        ViewBag.Flights = flights;
        ViewBag.Dates = new SelectList(dates);
        ViewBag.Class = new SelectList(clss);
        DateTime dateTime;
        if (Date != null)
        {
            DateTime.TryParse(Date, out dateTime);
        }
        else
        {
            dateTime = DateTime.Now;
        }

        if (ModelState.IsValid)
        {
            TicketSale ticket = new TicketSale();
            ticket.FirstName = model.FirstName;
            ticket.LastName = model.LastName;
            ticket.PassportSeries = model.PassportSeries;
            ticket.PassportNum = model.PassportNum;

```

					ДР.122.4142.03.ПЗ	Аркуш
						116
Зм.	Аркуш	№ документа	Підпис	Дата		

```

        ticket.Age = model.Age;
        ticket.PhoneNum = model.PhoneNum;
        ticket.Email = model.Email;
        ticket.Class = model.Class;
        ticket.Registered = false;

        var dp = db.Departures.Include(p => p.Flight).Include(p =>
p.Crew).Include(p => p.Aircraft).Include(p => p.AirportTax).Include(p =>
p.Rate);

        if (dp.SingleOrDefault(p => p.Flight.FlightName ==
FlightName && p.DepartureDate == dateTime) == null)
        {
            ModelState.AddModelError("", "There are no flights on
this date");

            return View("FlightError");

        }
        ticket.DepartureId = dp.Single(p => p.Flight.FlightName ==
FlightName && p.DepartureDate == dateTime).Id;
        ticket.Departure = dp.Single(p => p.Flight.FlightName ==
FlightName && p.DepartureDate == dateTime);

        double classCoff;
        if (ticket.Class == "Economy class") classCoff = 1;
        else if (ticket.Class == "Business class") classCoff = 3;
        else classCoff = 6;

        AirportTax airportTax = db.AirportTaxes.Single(p =>
p.AirportTaxName == ticket.Departure.AirportTax.AirportTaxName);
        Rate rate = db.Rates.Single(p => p.RateName ==
ticket.Departure.Rate.RateName);
        ticket.TicketPrice = (airportTax.amountOfMoney +
rate.amountOfMoney) * classCoff;
        Departure d = db.Departures.Find(ticket.DepartureId);
        d.NumOfPurchasedTickets++;
        db.TicketSales.Add(ticket);
        db.Entry(d).State = EntityState.Modified;
        db.SaveChanges();
        int id = db.TicketSales.Max(p => p.Id);
        TicketSale ticketSale = db.TicketSales.Find(id);
        return View("Res", ticketSale);
    }
    return View(model);
}
}
}

```

### Лістинг 3 – AircraftList.cshtml

```

@model IEnumerable<Airport_IS.Models.Entities.Aircraft>

@{
    ViewBag.Title = "AircraftList";
}

<h2>Aircraft list</h2>

```

					ДР.122.4142.03.ПЗ	Аркуш
						117
Зм.	Аркуш	№ документа	Підпис	Дата		

```

@using (Html.BeginForm("Create", "Aircraft", FormMethod.Get))
{
<fieldset class="form-inline">
    <p>
        <input type="submit" value="Create new" class="btn btn-success" />
    </p>
</fieldset>
}
<table class="table">
    <tr>
        <th>
            @Html.DisplayNameFor(model => model.AirCraftBrend)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Speed)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.NumberOfSeats)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Fuel)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.TailNum)
        </th>
    </tr>

    @foreach (var item in Model)
    {
        <tr>
            <td>
                @Html.DisplayFor(modelItem => item.AirCraftBrend)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Speed)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.NumberOfSeats)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Fuel)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.TailNum)
            </td>

            <td>
                @Html.ActionLink("Edit", "Edit", new { id = item.Id }, new
                { @class = "btn btn-success" })
                @Html.ActionLink("Delete", "Delete", new { id = item.Id },
                new { @class = "btn btn-success" })
            </td>
        </tr>
    }
</table>

```